"School of Science & Technology"
" Master in Information Systems (PLS)"

Postgraduate Dissertation
"Customer sentiment analysis"

Christina Katsiri

Supervisor: "Panagiotis Symeonidis"

Patras, Greece, "January" "2024"

"Customer sentiment analysis"

Christina Katsiri

Supervising Committee

Supervisor:
"Panagiotis Symeonidis"
"Affiliation"

Co-Supervisor:
"Georgia Gkarani"
"Affiliation"

Patras, Greece, "January" "2024"

"Acknowledgments and / or Dedication"

Abstract

The massive use of digital platforms and social media makes it necessary for almost all the companies to use and evaluate their customers reviews.

With this dissertation we tried to present an extensive study on the effectiveness of advanced Natural Language Processing (NLP)[1] techniques focusing on semantic similarity measures— WuPalmer[2] similarity, Path[3] similarity, Leacock-Chodorow (LCH)[4] similarity, and Resnik (RES)[5] similarity—alongside the VADER[6] sentiment analysis tool and the BERT[7] (Bidirectional Encoder Representations from Transformers) model. The research aims to identify the most effective technique for interpreting customer feedback across various diverse digital communication channels and provide to customers a useful tool through which they can extract important information for products or interest.

For the existing reviews analysis two models are examined, VADER[6] and BERT[7].

The VADER[6] model, a lexicon and rule-based system, is assessed for its precision in sentiment polarity detection, particularly in concise text responses. The BERT[7] model, a deep learning powerhouse, is examined for its advanced capability to contextualize sentiments in more elaborate and nuanced feedback.

A thorough theoretical exploration of the examined semantic similarity measures is followed. WuPalmer[2] similarity is analyzed for its ability to quantify the conceptual distance between words in customer feedback. Path[3] similarity is considered for its fundamental approach in measuring semantic distances. LCH[4] similarity, an information-content-based measure, is evaluated for its effectiveness in capturing the depth of taxonomic trees. RES[5] similarity, another information-content-based approach, is scrutinized for its unique perspective in considering the shared information between concepts.

Utilizing a comprehensive dataset of customer feedback for hotels in booking.com, the dissertation conducts a comparative analysis of these methodologies. This comparison not only seeks to determine the accuracy of each approach in classifying sentiments but also evaluates their performance concerning computational efficiency, scalability, and adaptability to varying feedback styles.

The research findings offer a critical insight into the advantages and limitations of each approach, providing a valuable resource for businesses and researchers in selecting the most fitting sentiment analysis tool for their needs. Furthermore, the study proposes an innovative hybrid model that integrates the strengths of semantic similarity measures and sentiment analysis tools, aiming to enhance the accuracy and reliability of customer sentiment analysis.

Keywords

Sentiment Analysis, Explainable AI, Reasoning Engine καιGenerative AI, visualisation

"Ανάλυση του συναισθήματος των πελατών για προϊόντα"

"Χριστίνα Κατσίρη"

Περίληψη

Η μαζική χρήση ψηφιακών πλατφορμών και κοινωνικών δικτύων καθιστά αναγκαία για σχεδόν όλες τις εταιρείες τη χρήση και την εκτίμηση των αξιολογήσεων των πελατών τους. Με αυτή τη διπλωματική προσπαθήσαμε να παρουσιάσουμε μια εκτενή μελέτη σχετικά με την αποτελεσματικότητα προηγμένων τεχνικών Επεξεργασίας Φυσικής Γλώσσας – Natural Language Processing (NLP[1]), εστιάζοντας στην χρήση semantic similarity measures όπως WuPalmer[2] similarity , Path[3] similarity, Leacock-Chodorow (LCH)[4] similarity και Resnik (RES)[5] similarity μαζί με χρήση των μοντέλων ανάλυσης συναισθημάτων VADER[6] και BERT[7] (Bidirectional Encoder Representations from Transformers). Η έρευνα στοχεύει λοιπόν στην εύρεση της πιο αποτελεσματικής τεχνικής για την ερμηνεία των ανατροφοδοτήσεων των πελατών μέσω διαφόρων ψηφιακών επικοινωνιακών καναλιών και την δημιουργία ενός εύχρηστου υσερ ιντερφαςε μέσω του οποίου οι χρήστες θα μπορούνε να εξαγάγουν σημαντικές πληροφορίες για προϊόντα ή υπηρεσίες που τους ενδιαφέρουν.

Για την ανάλυση των υφιστάμενων σχολίων των πελατών εξετάζονται δυο μοντέλα , το VADER[6] και το BERT[7]. Το μοντέλο VADER[6], ένα σύστημα βασισμένο σε κανόνες, αξιολογείται για την ακρίβειά του στην ανίχνευση της πολικότητας των συναισθημάτων, ιδιαίτερα σε συνοπτικές απαντήσεις κειμένου. Το μοντέλο BERT[7] χαρακτηρίζεται από την προηγμένη του ικανότητα να δίνει εννοιολογική έννοια στα συναισθήματα με πιο σύνθετη και λεπτομερή ανατροφοδότηση.

Στην συνέχεια η έρευνα παρουσιάζει μια θεωρητική ανάλυση κάθε μέτρου σημασιολογικής ομοιότητας. WuPalmer[2] similarity χαρακτηρίζεται από την ικανότητά της να ποσοτικοποιεί την εννοιολογική απόσταση μεταξύ των λέξεων . Path[3] similarity χαρακτηρίζεται από την ικανότητα στην προσέγγισή της μέτρηση σημασιολογικών αποστάσεων. Η LCH[4] similarity άλλη μια μέθοδος βασισμένη στο περιεχόμενο της πληροφορίας, αξιολογείται για την αποτελεσματικότητά της στην καταγραφή του βάθους των ταξινομικών δέντρων.Η RES[5] similarity, μια άλλη προσέγγιση βασισμένη στο περιεχόμενο πληροφορίας, χαρακτηρίζεται για τη μοναδική της προοπτική στην εξέταση της κοινόχρηστης πληροφορίας μεταξύ εννοιών.

Χρησιμοποιώντας δεδομένα με ανατροφοδοτήσεις πελατών στην ηλεκτρονική πλατφόρμα booking.com, η διατριβή διεξάγει μια συγκριτική ανάλυση των παραπάνω μεθοδολογιών. Αυτή

η σύγκριση δεν επιδιώκει μόνο να καθορίσει την ακρίβεια κάθε προσέγγισης στην κατηγοριοποίηση των συναισθημάτων αλλά και να αξιολογήσει την απόδοσή τους ως προς την υπολογιστική αποδοτικότητα, την επεκτασιμότητα, και την προσαρμοστικότητα σε διάφορα στυλ ανατροφοδότησης.

Τα ευρήματα της έρευνας προσφέρουν μια σημαντική εισαγωγή στα πλεονεκτήματα και τους περιορισμούς κάθε προσέγγισης, παρέχοντας μια πολύτιμη πηγή για επιχειρήσεις και ερευνητές στην επιλογή του πιο κατάλληλου εργαλείου ανάλυσης συναισθημάτων για τις ανάγκες τους. Επιπλέον, η μελέτη προτείνει ένα καινοτόμο υβριδικό μοντέλο που ενσωματώνει τα δυνατά σημεία των μέτρων σημασιολογικής ομοιότητας και των εργαλείων ανάλυσης συναισθημάτων, με στόχο την ενίσχυση της ακρίβειας και της αξιοπιστίας της ανάλυσης συναισθημάτων πελατών.

Συνοψίζοντας η διατριβή αυτή τονίζει τον κρίσιμο ρόλο της ανάλυσης συναισθημάτων στην ψηφιακή εποχή και προετοιμάζει το έδαφος για μελλοντική έρευνα στον τομέα εστιάζοντας στην ενσωμάτωση και τη βελτιστοποίηση των τεχνικών NLP[1] για βελτίωση ανατροφοδότησης πελατών.

Λέξεις Κλειδιά

Ανάλυση συναισθημάτων, Μηχανή συλλογισμού, γενετική τεχνητή νοημοσύνη, επεξηγήσιμη τεχνητή νοημοσύνη

Table of Contents

Περιεχόμενα

List of Figures

List of Tables

List of Abbreviations & Acronyms

1 Introduction

In the era of digital communication and e-commerce, understanding customer opinions has become crucial for businesses aiming to thrive in a competitive market but this could be also a good tool for consumers too. This dissertation explores the realm of sentiment analysis in customer reviews, a field that intersects natural language processing (NLP[1]), data analytics, and consumer behavior. The objective is to harness advanced computational techniques to decode the nuanced expressions of customer sentiments, providing valuable insights , mainly for consumers and researchers alike.

The study capitalizes on the power of Python, a versatile programming language renowned for its robust libraries and tools in data science and machine learning. Python's accessibility and efficiency make it an ideal choice for handling and analyzing large volumes of text data inherent in customer reviews.

Epicenter of this research are sophisticated NLP[1] tools , groundbreaking models such as VADER (Valence Aware Dictionary and sEntiment Reasoner)[6] and BERT[7] (Bidirectional Encoder Representations from Transformers). VADER[6], with its rule-based system optimized for social media text, is adept at handling short-form content, often seen in customer reviews. In contrast, BERT[7], a state-of-the-art machine learning model, excels in understanding the context of entire sentences and paragraphs, making it well-suited for in-depth sentiment analysis.

Furthermore, the dissertation uses similarity measures, including WuPalmer[2] similarity, Path[3] similarity, Leacock-Chodorow (LCH)[4] similarity, and Resnik (RES)[5] similarity. These measures are pivotal in understanding the semantic relationships and contextual nuances within customer feedback. By comparing and contrasting these measures, the study aims to uncover the most effective methods for interpreting the subtleties of customer sentiment.

This research not only delves into the theoretical aspects of these tools but also applies them practically to a dataset of customer reviews. Through this application, the dissertation seeks to evaluate the effectiveness, accuracy, and scalability of each method. The comparative analysis aims to provide a comprehensive understanding of how different NLP[1] techniques and models can be used in order to get the insights from customer reviews, thereby contributing to the fields of sentiment analysis, consumer research, and data analytics.

As consumers increasingly use digital communication and e-commerce, this study offers them a way to easily take advantage of existing customer reviews of any product of interest. The findings are expected to have significant implications for how consumers proceed

to product researched and eventually for how companies understand and respond to their customers needs, ultimately influencing decision-making processes and customer relationship management.

2 Related Work

2.1 Sentiment Analysis

Sentiment analysis, a pivotal aspect of natural language processing (NLP[1]), aims to identify and categorize opinions expressed in textual data. Originating from the broader concept of opinion mining, sentiment analysis has evolved to focus not only on extracting subjective information but also on perceiving the underlying emotional tone.

At its core, sentiment analysis is deeply rooted in linguistics, particularly in the fields of Pragmatics and Semantics. Pragmatics examines how context influences the interpretation of meaning in language, essential for understanding the often nuanced and context-dependent nature of sentiments in customer reviews. Semantics, dealing with the meaning conveyed by words and sentences, helps in identifying sentiment polarity. The Speech Act Theory is also pertinent, as it elucidates how phrases in customer reviews can be characterized as complaints or praises.

The psychological dimension of sentiment analysis is crucial in understanding how emotions are conveyed through language. Theories such as Paul Ekman's classification of basic emotions provide a framework for categorizing emotional expressions in text. Affective computing, an interdisciplinary field, plays a significant role in this aspect of sentiment analysis, aiming to develop systems capable of recognizing and interpreting human emotions.

The foundational concepts of polarity (positive, negative, neutral) and subjectivity (subjective opinions vs. objective statements) are integral to sentiment analysis. Early sentiment analysis work involved creating sentiment lexicons, which are comprehensive lists of words annotated with their respective sentiment polarities. However, these initial methods often struggled with the complexity and context-dependency of language in customer reviews.

Initially, sentiment analysis relied heavily on rule-based systems that used these lexicons. While effective to an extent, these systems were limited in their ability to grasp the context and the complex sentiment expressions have. The advent of machine learning marked a significant shift in sentiment analysis. Techniques such as Support Vector Machines (SVM) and later, neural network models, allowed for more sophisticated, nuanced sentiment analysis that took into account not just individual words, but the context and subtlety of entire phrases and sentences.

2.2 Sentiment Analysis Models

There are three available approaches that can be used in terms of sentiment analysis and can be found below:

2.2.1 Rule-based

This approach identifies, classifies, and scores specific keywords based on predefined rules called lexicons. Lexicons are compilations of words representing the writer's intent, emotion, and mood. Marketers assign sentiment scores to positive and negative lexicons to reflect the emotional weight of different expressions. To determine if a sentence is positive, negative, or neutral, the software scans for words listed in the lexicon and sums up the sentiment score. The final score is compared against the sentiment boundaries to determine the overall emotional bearing. The advantage of rule-based models is their transparency, but they can be limited in handling complex language nuances and contextual meanings.

2.2.2 Machine Learning (ML)

This approach is based on machine learning (ML) techniques and sentiment classification algorithms (neural networks and deep learning) in order to train computer software to emotionally identify sentiment from text. It includes creating a sentiment analysis model and training it repeatedly on known data so that sentiment of unknown data to be guessed with high accuracy. Currently there are a lot of pretrained models available. If the trained data are enough the accuracy is great giving huge advantages to this approach. However, we should never forget that a trained ML model is usually specified to one business area. This means that we have to select the trained model better suits to our needs.

2.2.3 Hybrid

Hybrid sentiment analysis works by combining different approaches like ML and rule-based or different types of machine learning models. Main scope is to use best features from all methods to optimize speed ,accuracy and make them suitable for specific applications. We must have in mind though that it takes time and technical efforts to successfully combine different systems.

In our case we studied a rule-based model (VADER[6]) and a machine learning model (BERT[7]) to find which of the two better fits to the data that we need to examine.

2.3 Sentiment Analysis Steps

Independently of the type of model used , sentimental analysis involves the below general steps.

2.3.1 Data Collection

First, we need to gather relevant brand reviews and mentions in one dataset. We can collect feedback from website or partner with resources that contains such data. The data can either be downloaded localy and then processed or we can use the online interface of the website.

- Data Processing

This step includes clean up the data by removing irrelevant information , for example removing special characters and correcting typos. It is useful to Part-of-Speech(POS) tagging words and also important to normalize text by converting to lower case, handling contractions, removing stop words and last lemmatize text into sentences or words.

- Use pre-trained models or model training

In this step we can either use a pre-trained model so as to analyze our processed data or we can train a model using a labeled dataset that will be separated to train and testing data.

- Post-processing and Interpretation

The results are analyzed and interpreted in the context of specific application or business need.

- Data visualization

In case there is an need to share the output of sentimental analysis it is important to turn them into graphs and charts for easier understanding. Something that is part of this research.

2.4 Generative AI in Sentiment Analysis

Generative AI refers to a subset of artificial intelligence that is capable of creating new content. This can include not only text but also images, audio, video, code, or other media. It is a broad term that covers a variety of machine learning systems, particularly those that can generate new data resembling the data they were trained on. The most common and known examples of GENAI are DALL-E , ChatGPT and Google Bard.

Last few years lot of papers related to Generative AI have been published. From our point of view the most interesting are

- Training language models to follow instructions with human feedback

This paper tries to prove that the precision and accuracy of a model is not depending only on their size. It was found out that a big language model may not follow user’s intent. To overcome this issue a new method is introduced. New models called InstructGPT were created. This models combine existing language models with human feedbacks , meaning that existing language models are again fine-tuned using human feedback.

It was found that the outputs of InstructGPT models were preferred in an excellent percentage contrary to existing bigger models.

Below you can find a picture describing the steps of InstructGPT model

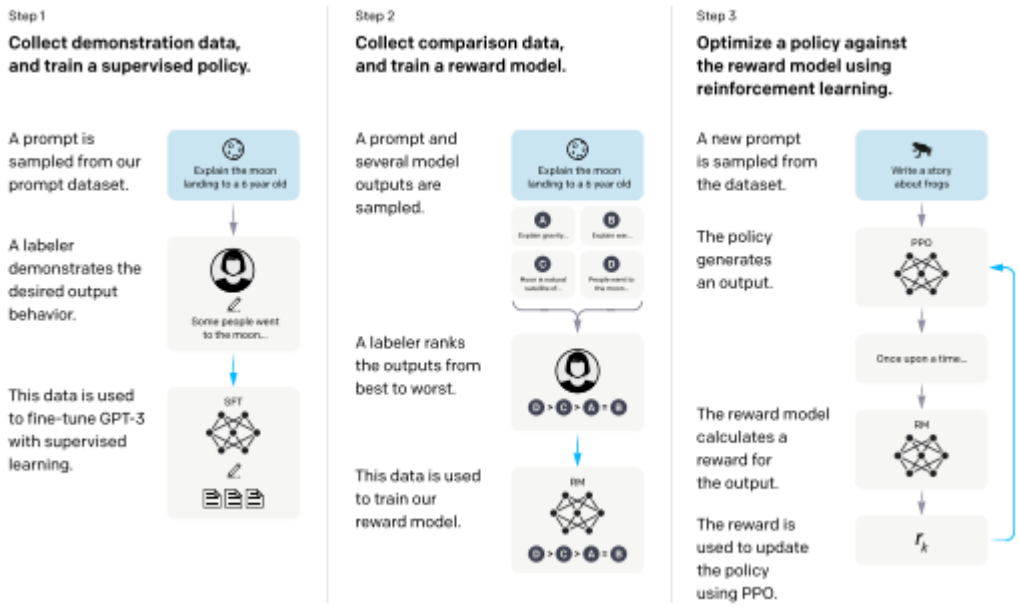

Figure 1 : Steps of InstructGPT model

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

This paper introduced a new deep learning model. BERT[7] stand for Bidirectional Encoder Representation from Transformers. BERT[7] is trained on a large corpus of text and learns to understand language patterns and sentiment from this data. BERT[7] model is bidirectional trained meaning that can learn information of a text from both left and right side. This gives a better and more accurate meaning on the context. Due to the above BERT[7] obtains excellent results in eleven tasks relevant to language processing.

2.5 Reasoning Engine in Sentiment Analysis

Reasoning engine refers to a system or part of an AI model that processes and interprets data to draw conclusions or make decisions. It’s designed to understand and analyze the sentiment of a text by considering various aspects like context, subtleties, and even sarcasm or irony. The Reasoning Engine applies logical rules or learned patterns to deduce the underlying sentiment in a given piece of text, distinguishing between positive, negative, or neutral sentiments. This is particularly useful in areas like customer feedback analysis, social media monitoring, and market research, where understanding public sentiment is crucial. There are few papers related to this subject and one of them is “Towards Reasoning in Large Language Models: A Survey”. This paper is presenting a detailed analysis of LLMs (Large Language Models) and their capability of reasoning.

There are also few mentions/ articles referring to differences and uses of reasoning Engine and search engine.

In our thesis a research engine becomes available to users so as to get specific and reasoning results on their research via a simple UI.

2.6 Explainable AI in Sentiment Analysis

Explainable AI (XAI) refers to artificial intelligence systems designed to make their operations understandable to humans. The main goal of XAI is to create AI models that are transparent and interpretable, allowing users to comprehend how and why the AI arrived at a particular decision or output. This is crucial in applications where trust and reliability

are important, such as healthcare, finance, or legal contexts. XAI techniques involve methods to visualize the decision-making process of AI or provide explanations for the model's predictions, making AI more accessible and accountable. One of the most interesting paper regarding explainable AI is the "Sentiment Analysis of Customer Reviews of Food Delivery Services Using Deep Learning and Explainable Artificial Intelligence: Systematic Review". This paper analyze the need of explainable AI as it was found that up to 77% of the models are not human friendly and uses cannot understand their value and output and thus cannot trust them.

In this paper we tried to cover the above needs by using explainable AI (plots , graphs , tables)

3 Similarity Measures , Train Models and Pre-Trained Models in Sentimental Analysis

3.1 Understanding Similarity Measures

Similarity measures are fundamental in various fields such as data analysis, machine learning and natural language processing (NLP[1]). Using this measures we are able to determine how similar two data objects are. To be more specific the measurement is performed between synsets, we usually get the first synset of each word as it is usually the most similar to the requested word.

The are several similarity measures and we will try to present the most common similarity measures in sentimental analysis

3.1.1 WuPalmer similarity

WuPalmer[2] similarity measures the similarity of two words by considering the depths of the synsets in WordNet along with the depth of LCS (Least Common Subsumer). LCS is the most specific ancestor node that both synsets share.

The similarity is calculated as the ratio of the depth of LCS to the sum of the depths of each synset to the LCS. Below you can find the mathematical type:

$$\text{WUP Similarity} = \frac{2 \times \text{depth}(\text{LCS}(\text{synset1}, \text{synset2}))}{\text{depth}(\text{synset1}) + \text{depth}(\text{synset2})} \tag{1}$$

The sysnset depths refer to the length of the path from the given synset to the root of the taxonomy (is our case WordNet)

The score of WuPalmer[2] similarity can be

$$0 < \text{score} \leq 1$$

where 1 indicates maximum similarity meaning that words are identical or have very close meaning because they share a common ancestor very close to them in the taxonomy.

While close to 0 indicates that the words are dissimilar. Sharing only a very distant common ancestor in the taxonomy. Note that score can not get 0 value as depth of LCS can never be zero.

WuPalmer[2] similarity is useful in tasks that require understanding the semantic relatedness of terms, although it has limitations in terms of context sensitivity and dependence on the specific taxonomy used as it does not consider the actual context in which words are used and is strictly depending on WordNet taxonomy.

3.1.2 Path similarity

Another measure of semantic similarity between two words is Path[3] similarity. It is based on a lexical taxonomy and more specific is quantifing how close these words are based on their shortest connecting path in the hypernym/hyponym taxonomy. The hypernym/hyponym taxonomy is often represented as a graph in lexical databases like WordNet where nodes are synsets and edges represent hypernymy (an kind of "is-a" relationship). In the taxonomy a hypernym is a more general term and a hyponym is a more specific term.

The score of Path[3] similarity can be found with the below type

$$\text{Path similarity} = \frac{1}{\text{shortest path between synset1 and synset2} + 1} \tag{2}$$

The "1" in denominator prevents division by zero and ensures that similarity score can be between 0 and 1. Again score equals to 1 depicts maximum similarity.

From the above it is obvious that Path[3] similarity is a fundamental measure in computational linguistics for assessing the semantic closeness of words based on the shortest path connecting them in a lexical taxonomy. Although it's useful for basic semantic assessments, its applicability can be limited by the structure of the taxonomy and the lack of context consideration.

3.1.3 Resnik similarity

Resnik (RES)[5] similarity is a method for measuring semantic similarity between two words based on information content. It is mainly used in natural language processing (NLP[1]) and computational linguistics, usually within frameworks like WordNet. To be more specific the Resnik (RES)[5] similarity calculates the semantic similarity based on notion of information content (IC). The core idea is that two words are more similar if they share a highly informative common ancestor in a taxonomy.

Resnik[5] similarity is defined as the information content of the Lowest Common Subsumer (LCS) of the two concepts , focusing solely on the shared information content. The formula can be found below

$$\text{Resnik Similarity} = \text{IC}(\text{LCS}(c1, c2)) \tag{3}$$

Where c_1 and c_2 are the concepts , LCS is the least common subsumer and IC is the information content.

Resnik (RES)[5] similarity is a powerful tool in computational linguistics for quantifying semantic similarity based on shared information content. Its reliance on the concept of information content offers a unique perspective compared to other similarity measures, though it also introduces specific limitations and dependencies, particularly related to the choice of corpus.

3.1.4 Leacock-Chodorow similarity

Leacock-Chodorow (LCH)[4] similarity is another metric that determines semantic similarity of two words depending on taxonomy like WordNet. LCH[4] is based on the shortest path that connects two synsets in a hypernym/hyponym taxonomy and scales this value by the maximum depth of the taxonomy. It is consider both distance and depth of the taxonomy and the mathematical type can be found below

$$\text{LCH Similarity} = -\log\left(\frac{\text{shortest path length between synset1 and synset2}}{2 \times \text{max depth of the taxonomy}}\right) \quad (4)$$

Where 'shortest length between synset1 and sysnset2' is the number of edges in the shortest path between two synsets in the WordNet taxonomy and 'max depth of the taxonomy' is the maximum depth of the taxonomy. It is used to normalize the path length , ensuring the similarity values are scaled within a standard range.

This normalization by the maximum depth allows a more nuanced similarity measurement compared to simpler path measurements.

The score of LCH[4] similarity can be

$$0 \leq \text{score} \leq 1$$

where 1 indicates maximum similarity and 0 indicates no similarity.

Concluding the Leacock-Chodorow similarity is a sophisticated measure that incorporates both the distance between words in a taxonomy and the depth of that taxonomy, offering more detailed and quiddity similarities. Its effectiveness in capturing subtle semantic relationships makes it valuable in various NLP[1] tasks, although its performance is highly dependent on the specific characteristics of the taxonomy employed.

3.2 Deep Dive into Pre-Trained Models

A pre-trained model is a model that has been trained on a large usually general dataset. The trained process involves various learning features such as learning language structures, grammar, context, and word relationships from large text corpora.

Regarding NLP[1] pre-trained models are usually used in tasks like sentiment analysis, question-answering , language translation etc. Pre-trained models are vital in NLP[1] as

training from scratch is often impractical due to complexity and size of language data that should be used.

There are several pre-trained models available , we will present few of them in next subchapters

3.2.1 BERT (Bidirectional Encoder Representations from Transformers)

BERT[7] is a groundbreaking model introduced by researchers at Google in 2018. It is trained in a huge not annotated text and unlike previous models that processed text in one directions BERT[7] understands the context of a word based on all of its surroundings (both left and right). BERT[7] is also based to transformer architecture, a attention mechanism that learns contextual relations between words or sub-words in a text and can be easily fine-tuned by adding a task-specific layer and a small dataset.

For all the above BERT[7] can be used in a range of NLP[1] tasks with remarkable results. We should not though forget to mention that it requires significant computational power and memory, and it is more complex than other traditional models

3.2.2 GPT-2 (Generative Pretrained Transformer 2)

GPT-2 is an advanced language processing neural network developed by OpenAI at 2019.It is trained on a huge dataset in a unsupervised manner and can generate coherent and contextually relevant text over several paragraphs.

It is another transformer-based model meaning that is consumes self-attention mechanisms to handle input texts

It can also performed a range of NLP[1] tasks such as translation, summarization, question answering while it can proceed to certain tasks without extra training. One of the important potential of GTP-2 is its ability to generate human like text making it a powerful tool especially for application like text summarization, content generation etc.

3.3 VADER (Valance Aware Dictionary and Sentiment Reasoner)

VADER[6] is a lexicon-based sentiment analysis tool , meaning that it uses a predefined list of words along with rules to analyze the sentiment of a given text. It was specifically

designed for extracting sentiment from social media texts, online language, including slang, emoticons, and abbreviations.

Unlike other models, VADER[6] easily deployed and does not require special hw resources. Moreover, can analyze text data in real-time making it suitable for applications where immediate sentiment analysis is needed. For each text that analyzes it provides positive, negative, neutral, and compound scores. Thus VADER[6] might be not as sophisticated as other models but offers a straightforward and effective solution for many applications with less requirements.

3.4 Application of Similarity Measures in Sentiment Analysis

In the realm of sentiment analysis , understanding the treasure of languages and the subtleties of human emotion requires sophisticated computational techniques. Among these similarity measures play a crucial role giving algorithms the opportunity to discern sentiment by compering texts to known sentiment-bearing documents. We will try to introduce the integration of similarity measures into sentiment analysis workflow, highlighting their impact through specific case studies and examples.

So in sentimental analysis the similarity measures help in several key areas such as:

- feature engineering, by quantifying the similarity between the vocabulary of a given text and a predefined set of positive and negative words, similarity measures can be used to generate features that are indicative of sentiment.
- context understanding, sentiment is often depending on the context in which words are used. Similarity measures can help algorithms to understand context by comparing sentence structures mad meanings to known patterns associated by specific sentiments.
- disambiguation, words with multiple meaning can lead to misinterpretation of sentiment. Similarity measures help in word sense disambiguation by comparing the context in which a word us used to the contexts of its various meanings.
- dimensionality reduction, by grouping words or phrases that are sentimental similar, similarity measures can reduce the complexity of text data, focusing analysis on the most sentiment-relevant aspects.

To be more practical similarity measures can be used in the below areas:

- product review analysis, there are a lot of studies that use similarity measures for analyzing product reviews by identifying semantically similar phrases across reviews. Using for example WuPalmer[2] and Path[3] similarity measures, the extraction of common themes and sentiments is enabled improving the accuracy of sentiment classification.

- social media monitor, a social media analytics platform integrated Resnik (RES)[5] similarity into their sentiment analysis model to better understand the context of slang and abbreviations commonly used in social posts. This integration lead to a more accurate assessment if user sentiments towards products and brands, facilitating more responsive customer service and brand management.

- customer feedback system, there is a customer feedback system that incorporates LCH[4] similarity measure. By this incorporation the feedback were more effectively categorized into positive, negative and neutral. By understanding the semantic depth of customer feedback in relation to service features, the tool provided actionable insights into areas requiring improvement.

- market research, in a market research project sentiment analysis was improved by using similarity measures to compare respondent opinions with a database of known sentiment expressions. This method streamlined the analysis of open-ended survey responses, uncovering nuanced sentiments toward various products and services.

3.5 Role of Pre-Trained Models in Enhancing Sentiment Analysis

One of the most significant advantages of pre-trained models is their ability to understand the nuances of languages and context. We will demonstrate same examples:

- contextual understanding, BERT[7]’s attention mechanism allows it to consider the entire context of a sentence or passage, enabling it to discern the sentiment of homonyms based on usage.

- sarcasm detection, the sequential nature of GPT-2 training enables it to follow narrative and tonal shifts within text, making it adept at identifying sarcasm, a traditional challenging area of sentiment analysis.

- domain specific sentiment, pre-trained models can be fine-tuned on domain specific databases, such as hotel review, to adapt their understanding of sentiment to specific vocabulary and expressions used in that domain.

We will try again to present some important case studies highlight the effectiveness of pre-trained models in sentiment analysis.

- hotel review sentiment analysis, a study utilized BERT[7] to analyze hotel reviews, fine-tuning the model with thousands of reviews to understand sentiment specific to the hospital industry. The model successfully identified nuanced sentiments related to different aspects of hotel service, such as cleanliness, staff friendliness and room comfort.

- social media sentiment monitoring, GPT-2 was employed to generate responses to social media posts, which were then analyzed for sentiment. This approach provided insights into public sentiment on various topics, demonstrating the model’s ability to understand context and nuanced language use.

3.6 Challenges and Considerations

3.6.1 Challenges in Implementing Similarity Measures and Pre-trained Models

Selecting to use pre-trained models or similarity measures has some issues that you should always have in mind. For example, advanced models like BERT[7] and GPT-2 demand significant computational resources, including high-end GPUs for training and inference. This requirement can limit accessibility for individuals and organizations with constrained budgets, potentially hindering research progress and application scalability.

Another important thing that must be considered is that both similarity measures and pre-trained models may amplify biases present in their training data. For instance, models trained predominantly on data from certain demographics may perform poorly on text from other groups, perpetuating stereotypes or under representing minority voices.

Furthermore ,while strides have been made in developing models for a variety of languages, the majority of pre-trained models are still most effective with English. This presents a challenge in analyzing sentiment in languages with fewer resources, dialects, or languages that rely heavily on contextual and cultural nuances.

At last sentiment analysis in specialized fields (e.g., medical, legal) requires models to understand domain-specific terminology and expressions. Fine-tuning pre-trained models with domain-specific data can be challenging, requiring substantial datasets that may not always be readily available.

3.6.2 Considerations for Model and Measure Selection

When selecting models and similarity measures for sentiment analysis, several considerations should guide the decision-making process. The choice of model or measure should directly align with the research or application goals. For instance, if detecting nuanced sentiment in customer feedback is the objective, a model known for its deep contextual understanding might be preferable.

We should not forget that the nature of the dataset—its size, language, domain, and annotation quality—should influence model selection. Larger datasets might allow for the use of more complex models, while smaller or specific datasets might be better from targeted fine-tuning or bespoke similarity measures.

Needless to say that available computational resources can significantly constrain model choices. It's essential to balance the desired model complexity with the practicalities of training and deployment environments.

Another aspect that should be carefully taken under consideration is the ability of models to provide transparency, interpretability and lack of bias. So we should be careful when selecting the dataset, we should know the possible application's impact and try to mitigate bias.

Concluding, the implementation of similarity measures and pre-trained models in sentiment analysis is full of challenges that extends to many domains such as technical, ethical, and practical. Successfully handling of these challenges requires a thoughtful approach that considers the specific requirements of the task, the characteristics of the data, and the broader implications of the analysis. By carefully selecting and applying these tools, researchers and practitioners can take advantage of their power to use existing data and provide important information based on data sentiment analysis.

4 Research Methodology and Experimental Setup

4.1 Research Objectives

The primary objective of this research is to enhance the decision-making process for potential hotel guests by providing an intuitive and informative platform that aggregates and analyzes hotel reviews. This goal is achieved through the development of a user interface that not only presents raw review data but also offers comprehensive sentiment analysis and comparative insights. The specific objectives include:

- Develop an Interactive UI: Design and implement a user-friendly interface that enables users to efficiently search for hotels and access detailed analyses of customer reviews.
- Apply Sentiment Analysis and Present Comparative Insights: Utilize mainly pre-trained model to analyze hotel reviews for sentiment, offering users insights into the overall trends and specific predefined aspects of each hotel. It also provides users with comparative insights such as sentiment scores, themes of positive and negative reviews, and overall guest satisfaction.
- Enhance User Decision Making: By aggregating and analyzing review data, the platform aims to assist users in making decisions about their hotel they finally choose, grounded in comprehensive data analysis.

The research objectives directly align with the methodologies discussed in the previous chapters by leveraging the power of NLP[1] and machine learning to process and analyze large datasets of hotel reviews. The development of an interactive UI that utilizes pre-trained model for sentiment analysis, represents a practical application of these methodologies, aiming to provide end-users with actionable insights derived from complex data analyses. This approach not only showcases the potential of these technologies in real-world applications but also addresses the need for tools that can extract meaningful patterns and sentiments from vast amounts of review.

4.2 Data Collection and Dataset Description

The primary source of our dataset is booking.com , a leading digital travel platform that offers a wide range of accommodation options. The platform allows users to post reviews

and rating based on their lodging experiences, providing a rich source of data on customer satisfactions and preferences. Booking.com exposes an API which can be used to get the needed info. This though requires a subscription so we based our research on an available in internet csv file that contain a huge amount of hotel reviews in booking.com.

The dataset contains half million of reviews structured in a csv comma separated format. It includes reviews for several hotel, having for each review info like hotel name, location, date of the review, reviewer's score , reviewer's nationality, hotel's average score and of course positive and negative review text.

Needless to mentioned that the dataset covers a wide range of accommodations from various geographical location, from budget hostels to luxury resorts, encompassing a broad spectrum of customer experiences and sentiments.

We should also state here that for testing purposes we have created two subsets of reviews, which were manually annotated to serve as training sets. Each subset consist of 60 reviews with all the information as in the main dataset.

4.3 Feature Extraction and Data Preparation

After concluding in the dataset that will be used for training/test purposes along with the main test data that will be used eventually in final presentation, we must proceed to data preparation part. The below actions/steps performed to as to be able to get basic statistic measurements from our datasets.

- Lowercasing: it is important as it standardize the dataset , simplifies further processing steps , reduce the feature space , improves matching accuracy and enhances model performance. It our case we use the .lower() basic Python string method.

- Tokenization and punctuation removal, the sentence is split is smaller parts ,such as words , and punctuation are removed. This step is vital as it reduce noise and extract meaningful features. We use .split(“ ”) method and string.punctuation() that contains various punctuation marks.

- Removing numbers and stop words , this enhances the focus on sentiment relevant content reduces computational complexity and also improves accuracy and efficiency of sentiment analysis models. In our code we utilize .isdigit() and stopwords.words() for any language of interest.

- Lemmatization, it is another very important step as it keeps the words in their base form (lemma) reducing the variability if the input data. It also ensuring that words are analyzed in their most meaningful form improving the accuracy and efficiency of various NLP[1] models. In this step we used pos_tag function (indicating whether a word is noun, verd ,adjective etc) , WordNetLemmatizer().lemmatize() method of NLTK library enabling more accurate lemmatization.

- Cleaning finalization , is the last step of data prepossessing and it bridges the gap between row data and data ready for NLP[1]. It enhances the quality and relevance of the data, and ensures that analysis is based on the most accurate and representative information available.

4.4 Model Selection and performance evaluation

In this research we decide to compare the performance of the similarity measures mentioned in chapter 3.1 along with BERT[7] and VADER[6] mentioned in chapter 3.2.

Thus we use the pre-annotated dataset of 60 reviews, which has 20 positive, 20 negative and 20 neutral reviews. For all the above methods we followed the process described in chapter 4.3 to prepossess the data so as to improve the accuracy of methods sentiment analysis.

The performance evaluation measures that we decided to use are Precision, Recall. For easier understanding of the results we will use also the precision-recall and ROC curve graphs.

- false positive rate (FPR)[8], is a metric that measures the proportion of false positive predictions out of all actual negative instances in the dataset. It represents the model's tendency to incorrectly classify negative instances as positive and the formula that calculates it is

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{5}$$

- Precision , is a metric used mainly in binary classification task to measure the accuracy of positive predictions made by a model. It quantifies the proportion of the true positive predictions among all the positive predictions made by the model. In other words , precision answers the question : 'Of all the instances predicted by the model as positive , how many were actually positive?'

  The formula through which we can calculate precision is

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6}$$

where
True Positives (TP) are the instances correctly classified as positive by the model.
and
False Positives (FP) are the instances incorrectly classified as positive by the model.
A high value of Precision indicates that the model has a low rate of false positive predictions

- Recall or true positive rate (TPR)[8], is another metric in binary classification task that measures the ability of a model to correctly identify all positive instances in the dataset. It qualifies the proportion of true positive predictions that were correctly identified by the model out of all actual positive instances in the dataset. In other words the recall answers the question: 'Of all the actual positive instances in the dataset , how many did the model correctly identified as positive.'

The recall score is calculated by the formula

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{7}$$

- Precision-recall curve, It plots the trade-off between the precision and recall of the model as the decision threshold varies. The area under the PR[8] curve (AUC-PR) summarizes the overall performance of the model across all thresholds. A higher AUC-PR value indicates better performance, with a perfect classifier having an AUC-PR equals to 1.0.

- ROC[8] curve, is is a graphical plot that illustrates the diagnostic ability of a binary classification model across various decision thresholds. It visualizes the trade-off between the true positive rate (TPR)[8] and the false positive rate (FPR)[8] as the classification threshold changes.

We started with the 4 similarity measures, each of them was trained based on our test dataset, taking as test size the 0.2 of the total dataset and threshold set to 20. We performed the same twice using different datasets at a time. In the below pictures we can find the Precision – Recall curve and ROC[8] curve for all the similarity measures in each test dataset.

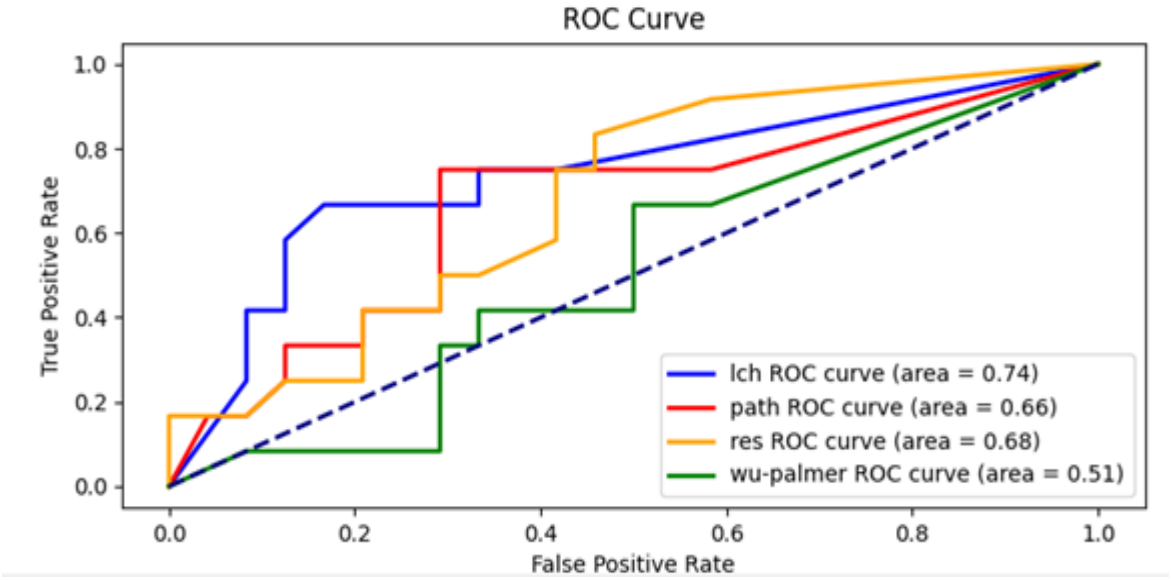

Figure 2 : Similarity measures ROC curve first test dataset

As already mentioned the ROC[8] curve illustrates the performance of a classification model across different thresholds. In the specific ROC[8] curve we can see that LCH[4] , PATH[3] and RES[5] starts with a sufficient discrimination between positive and negative classes which is continuously increasing. On the other hand WuPalmer[2] seems to rather have a essentially random guessing as is AUC[9] values are very close to area under the curve limits. LCH[4] similarity measure has the best AUC[9] value for the specific test data.

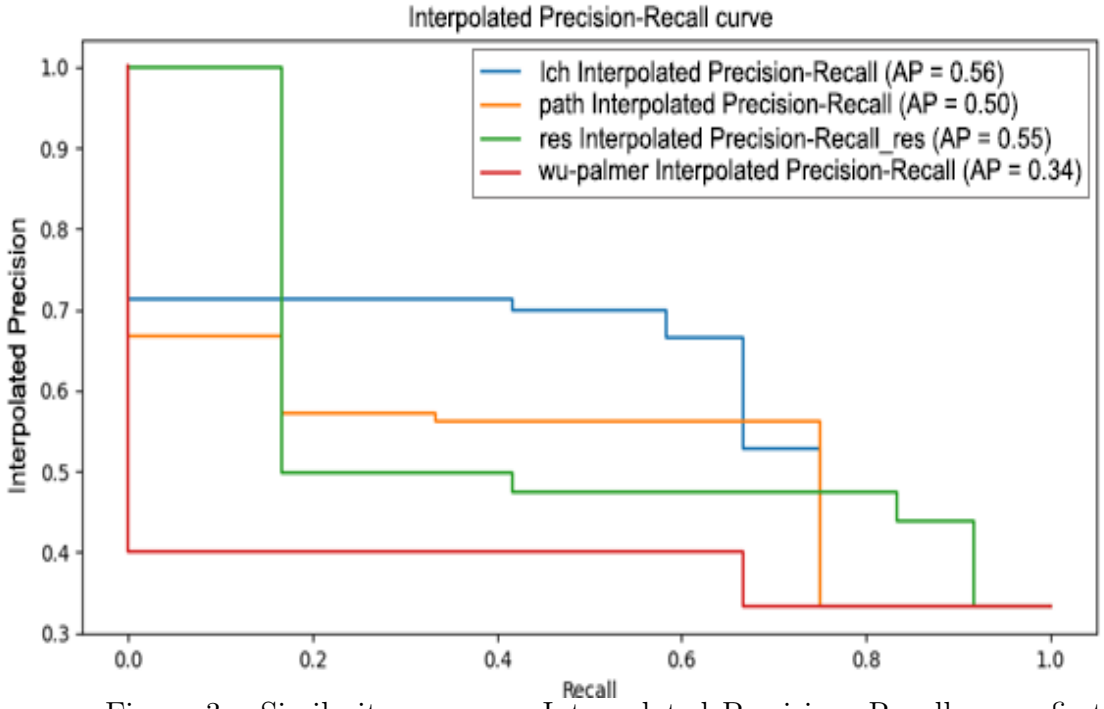

Figure 3 : Similarity measures Interpolated Precision- Recall curve first test dataset

The above figure illustrates the relationship between interpolated precision and recall of the interesting models in several threshold values. It is obvious that LCH[4] similarity measures has a more stable performance in comparison with other measures. We can notice also that RES[5] has the best performance but it did not last for the majority of thresholds thus we can not mark it as the best similarity measure. Also we have gathered in the below tables the values of the metrics for specific threshold values. The below values are in totally identification with the conclusion we made for ROC[8] and interpolated Precision-Recall curves. In our case where that is important to minimize false positives the LCH[4] similarity measure is the one that best feats. LCH[4] has the highest precision almost in all threshold values. Let us mention that thresholds in a precision-recall curve represent the values above which a model will predict that an example belongs to a specific class (usually the positive class), while below these values, it will predict it does not belong to that class (usually the negative class). Each threshold generates a point on the precision-recall curve, defining a specific value for both precision and recall. This happens for different threshold values.

In practice, thresholds affect how examples are classified based on the model's predictions. This means that you can adjust the threshold to predict more or fewer positive examples, depending on how conservative or demanding you are in your predictions. Precision-recall curves help us understand how these adjustments affect the performance of our model. In our case it is obvious that we need accuracy and not the positive prediction amount.

Threshold	Precision WuPalmer	Recall WuPalmer	Precision LCH	Recall LCH	Precision RES	Recall RES	Precision Path	Recall Path
0.16	0.435	0.833	0.555	0.833	0.400	0.833	0.526	0.833
0.26	0.500	0.666	0.600	0.750	0.571	0.666	0.642	0.750
0.75	0.571	0.333	0.700	0.583	0.625	0.416	0.727	0.666
0.95	0.666	0.166	0.666	0.333	0.666	0.166	0.600	0.250

Table 1 : Performance metric for first dataset trained models

We followed with the same analyses using an other test data set. Below we can find the relevant graphs and table data.

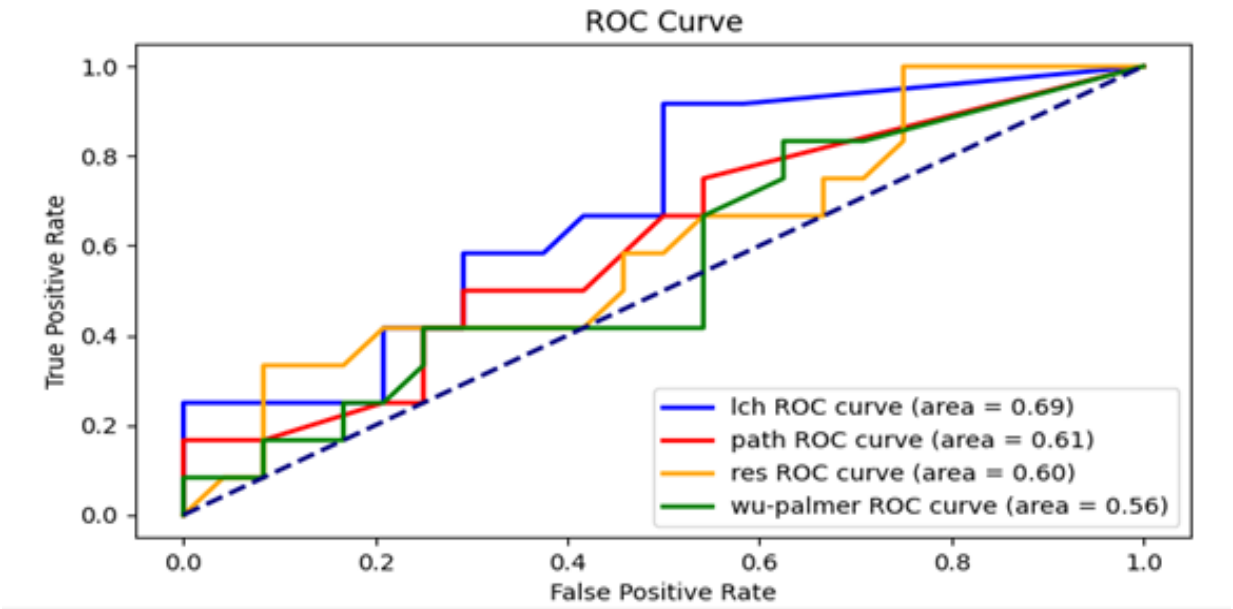

Figure 4 : Similarity measures ROC curve second test dataset

We can noticed that the performance of all similarity measures is almost the same and follows the same trend. Again LCH[4] seems to have the better AUC[8] value worst than the first test data set but again better than others. WuPalmer[2] seems to react better in this data set but again close to area under the curve limits.

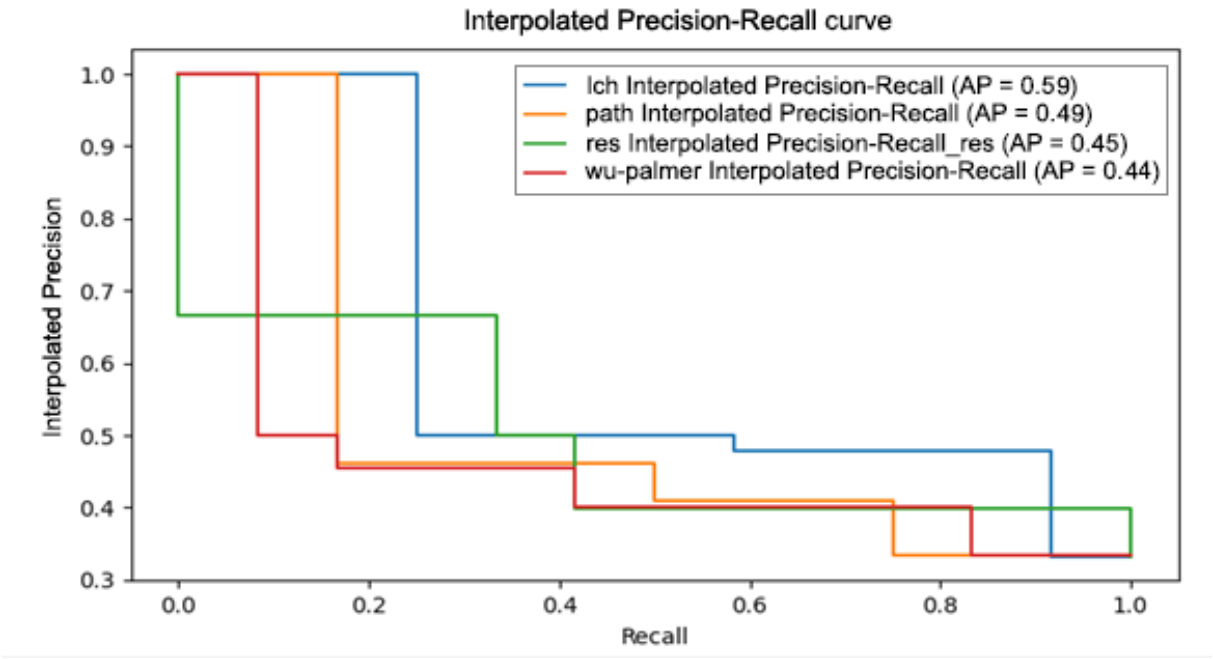

Figure 5 : Similarity measures Interpolated Precision-Recall curve second test dataset

From the above figure we can see that for this test data all measures except of RES[5] have the same high precision value. Again thought LCH[4] is the most stable leading us to believe that is it the best of our case

Lastly ,below we gathered the values of the metrics for specific threshold values. As we can see LCH[4] has again the best average precision and it is obvious that it has a better performance is highest thresholds

Threshold	Precision WuPalmer	Recall WuPalmer	Precision LCH	Recall LCH	Precision RES	Recall RES	Precision Path	Recall Path
0.16	0.550	0.916	0.625	0.833	0.444	0.666	0.500	0.750
0.26	0.562	0.750	0.615	0.666	0.437	0.583	0.562	0.750
0.75	0.555	0.416	0.583	0.583	0.444	0.333	0.666	0.500
0.95	0.750	0.250	0.833	0.416	1.00	0.083	0.600	0.250

Table 2 : Performance metric for first dataset trained models

From the above table we can see that precision-recall values varies based to threshold. Although taking into consideration the average value of precision-recall and ROC[8] curve of each measure, LCH[4] seems to be better for our dataset.

We continued our test by evaluating pre-trained model BERT[7] and VADER[6]. We again used two different test data set and again we changed the thresholds based on which we categorize the reviews to the classifiers.

Below we can find the precision-recall and ROC[8] curve graphs from first date set. It seems that both VADER[6] and BERT[7] have a more stable behavior. Comparing the two of them it seems that VADER[6] has a better accuracy in sentiment analysis , predicted better and predefined classifiers.

In the below pictures you can find the precision-recall and ROC[8] curve graphs for the two methods using each time a different test dataset.

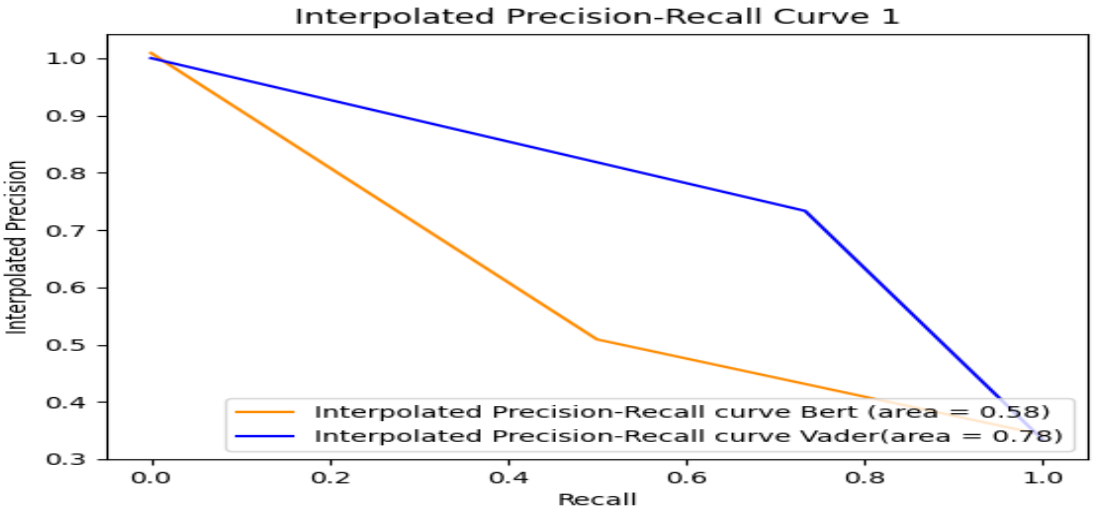

Figure 6 : Interpolated Precision-Recall Curve VADER vs BERT first test data

In figures 6 and 7 we can check the relationship between interpolated precision and recall of VADER[6] and BERT[7] using different test data along with ROC[8] curve. As we can see VADER[6] method have a better for our case proportion. oth methods are always over the area under the curve something very promising for both models performance. Let us mention again that it is important to have an accurate model , meaning that we prefer models with high precision values combined with recall at reasonably high levels.

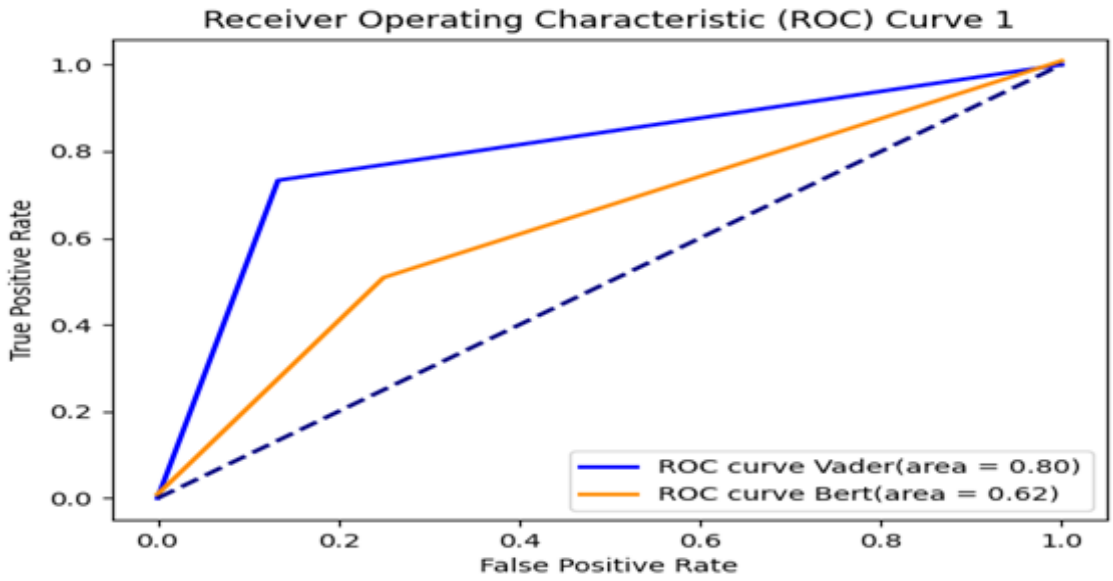

Figure 7 : Similarity measures ROC curve VADER vs BERT first test data

In the table below we gather the outputs of our test,using the already mentioned first dataset while setting different thresholds. It seems that both models are not significant affecting by the threshold modification.

Precision Vader	Recall Vader	Precision Bert	Recall Bert
0.333	1	0.333	1
0.733	0.600	0.483	0.455
0.75	0.700	0.500	0.490
1	0	1	0

Table 3 : Performance metric for first dataset pre-trained models

We continued with the second data set again with the same ,as the above experiment, threshold modifications. From the below figures 8 and 9 , we can see that both models keep the same trends with a small reduction on VADER[6] precision values.

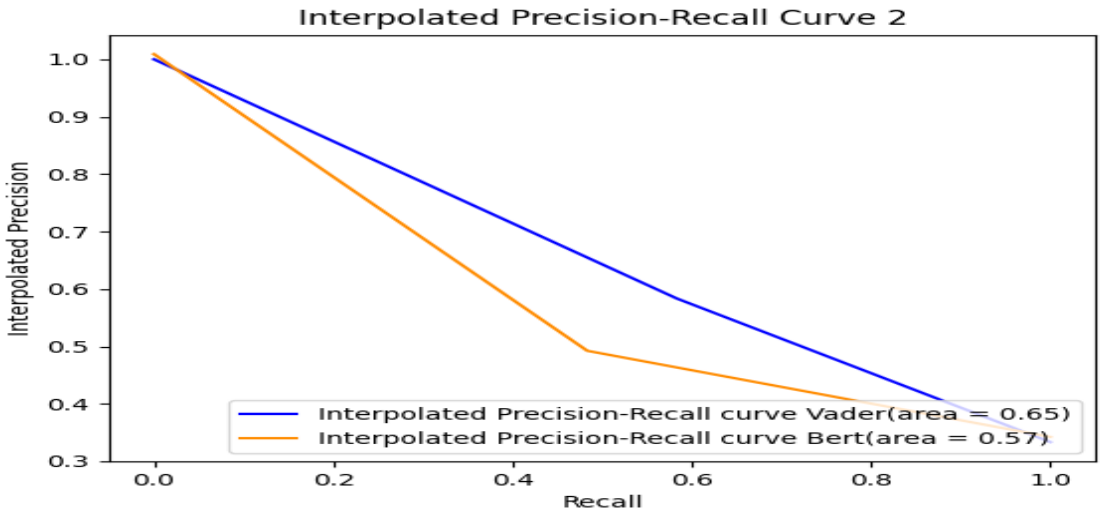

Figure 8 : Interpolated Precision-Recall Curve VADER vs BERT second test data

From ROC[8] curve illustrated in figures 9, we can see that again both VADER[6] and BERT[7] are always over the area under the curve. Which means that both models have an adequate ability to discriminate between positive and negative instances.

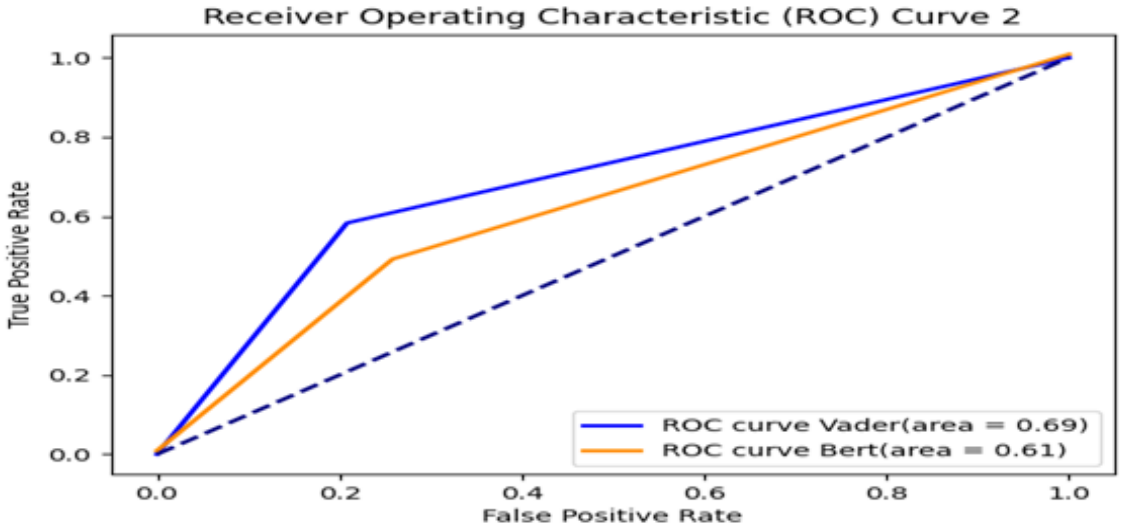

Figure 9 : Precision-Recall Curve VADER vs BERT second test data

In the table 4 we can find the values precision-recall got for both models in different threshold limits. Again precision of VADER[6] is higher than BERT[7].

Precision Vader	Recall Vader	Precision Bert	Recall Bert
0.333	1	0.333	1
0.566	0.500	0.483	0.450
0.583	0.550	0.516	0.500
1	0	1	0

Table 4 : Performance metric for second dataset pre-trained models

As we have though to select one of the two models it is obvious that VADER[6] has better performance as it is able to predict better both positive and negative instances than BERT[7]. In both test data the ratio between TPR[8] and FPR[8] is always bigger in VADER[6].

For all the above we concluded in using VADER[6] for our implementation even if this is opposite to our initial thoughts. The below images where we gathered the metrics of all models confirms our decision.

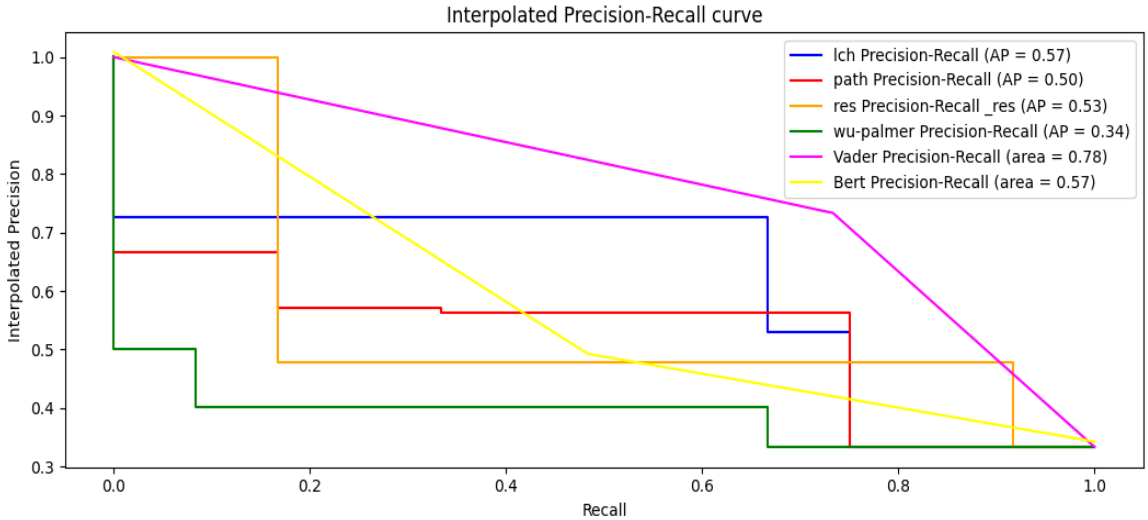

Figure 10 : Precision-Recall Curve first test data

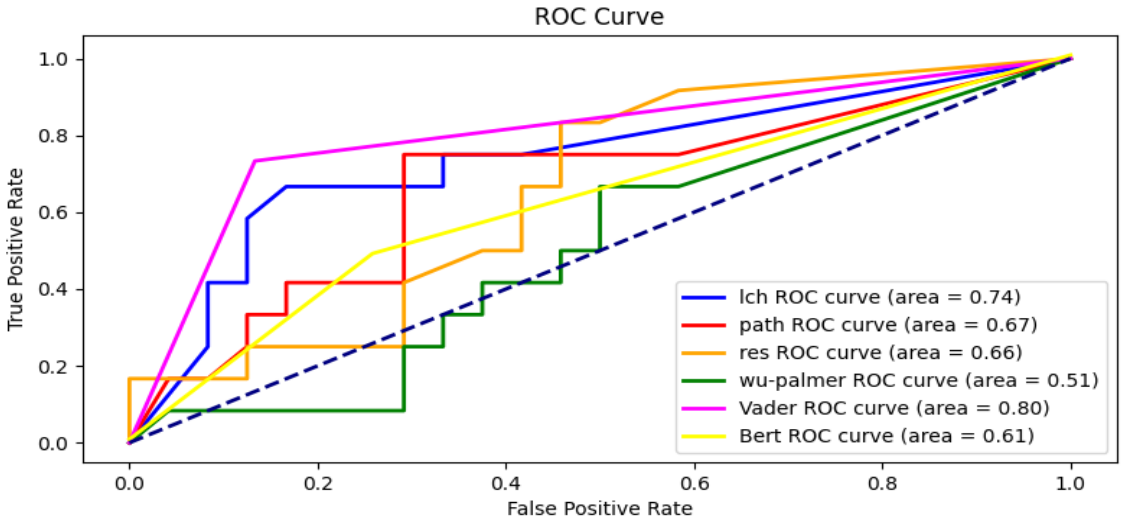

Figure 11 : ROC Curve first test data

Both ROC and Interpolated Precision-Recall curve of first test data illustrates the Vader’s better performance. Curves of second test data give also the same conclusion something that strengthen our decision.

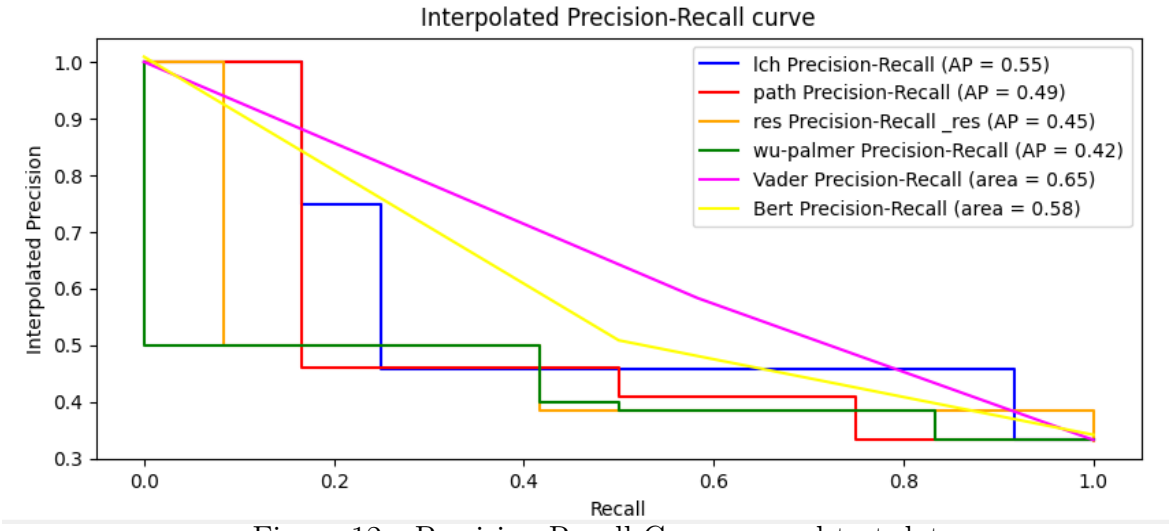

Figure 12 : Precision-Recall Curve second test data

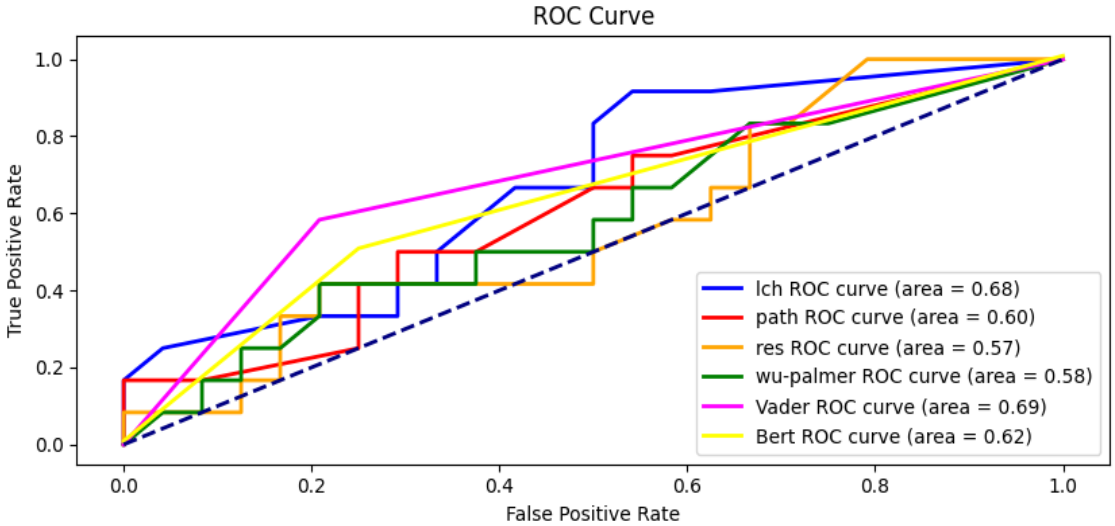

Figure 13 : ROC Curve second test data

Based on theoretical inputs we believed that the better decision was using BERT[7] for sentiment analysis. New pre-trained models offering superior performance and flexibility especially for complete NLP[1] tasks. It utilizes bidirectional contest and deep contextual embedding to capture complex linguistic relationships.

On the other hand, BERT[7] requires significant computation resources (CPUs and memory) as there are large and complex models with hundreds of millions or even billions of parameters. In cases of luck of needed resources, it can be extremely slow. It is not easy to understand how BERT[7] arrives to each prediction and often difficult to fine tune it as it requires substantial amounts of task-specific labeled data to achieve optimal performance.

The tests output though leads us to using VADER[6] sentiment analysis tool. A rule-based sentiment that is fast and computationally efficient. It can process large amount of text data quickly, great for real-time applications and processing social media data. It can be adaptable to various languages and domain as it does not rely on pre-trained models based on specific domains, but it uses lexicon-based approaches. It is very efficient in words that are modifies by negation or intensifiers and in social media texts as is handles the nuance of informal language, emotion, acronyms etc.

Of course, VADER[6] has disadvantages too. As VADER[6] was primary designed for English language, there might be a degradation when using in other languages. Its performance can be influenced by the subjectivity and biases inherent in its lexicon and rule-based

approach, leading to different results based on the composition of its lexicon and the rules defined for sentiment analysis. Lastly fine-tuning VADER[6] for specific domains or tasks may require additional effort and expertise.

Concluding as VADER[6] offers a fast and efficient solution for sentiment analysis, particularly for social media text with disadvantages that believing that will not afflict the overall output of our work, we use this tool for the sentimental analysis of existing customer’s reviews.

4.5 System Implementation and User Interface

The artifact of this thesis is a UI that allows users to view important information about services of interest.

There is a free text where user is able to type part of the hotel’s name and the drop down menu will show only the hotels which name contains the text use inserts.

Figure 14 : User Interface main page

Then user must select the requested hotel.

Figure 15 : User Interface main page, select hotel name

After selecting the hotel name , user is able to use the next drop down menu “Select Action”

This menu has the below options

- view graphs
- view all reviews
- check reviews and tags
- view positive reviews
- view negative reviews
- view sum up of the reviews

the user must select one of the options and press selected action.

The UI has also a help button which gives detailed info for all the actions can be performed via it.

4.5.1 View graphs

The user interface allows users to view several graphs with useful visualized information about the hotel that was chosen. We should mention that the data which are used for extracting this information are the cleaned and sentiment analyzed reviews of the specific reviews using the methods described in previous chapters.

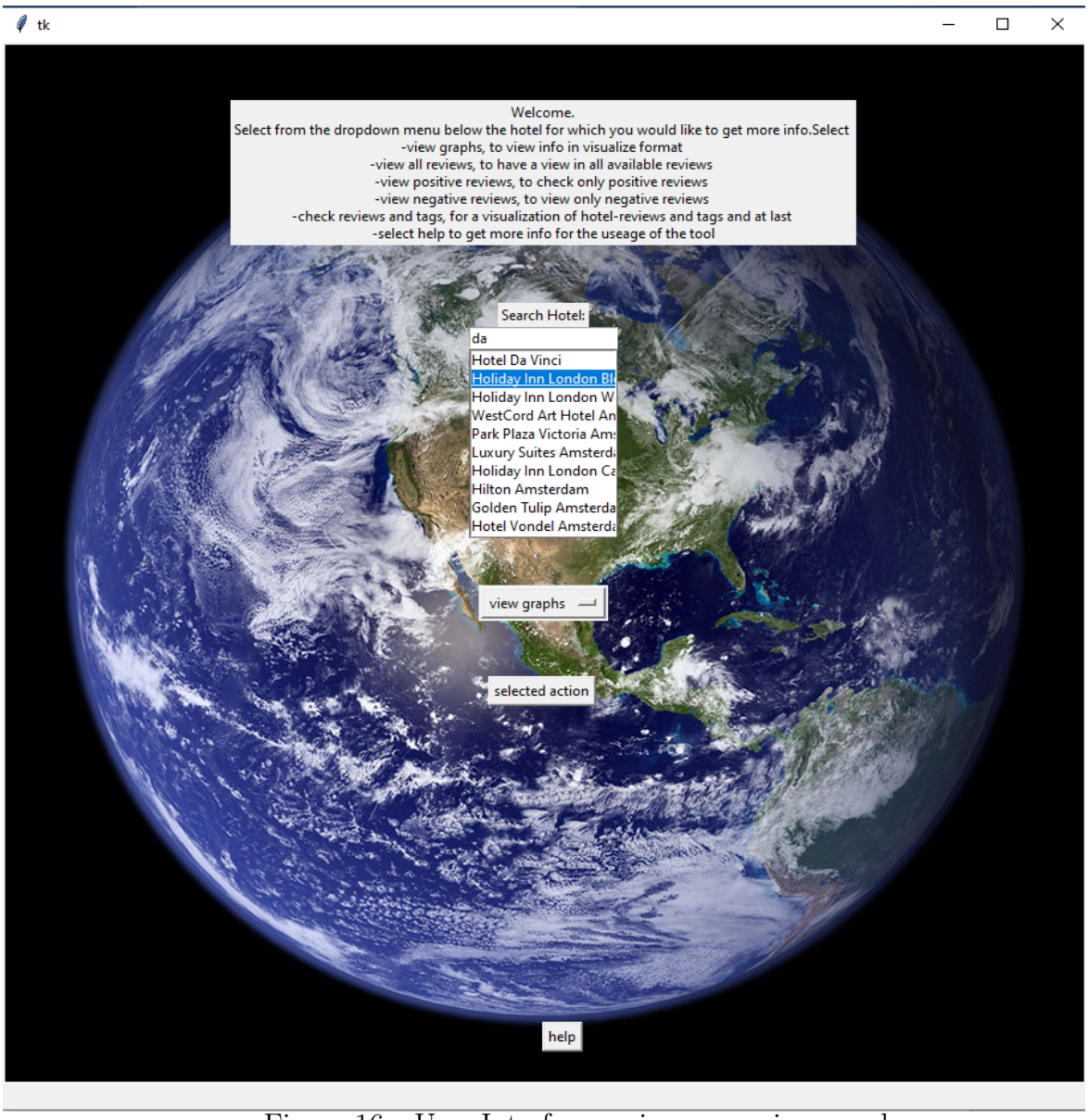

Figure 16 : User Interface main page , view graphs

The first picture that user sees is the number words of interest appears in the available reviews.

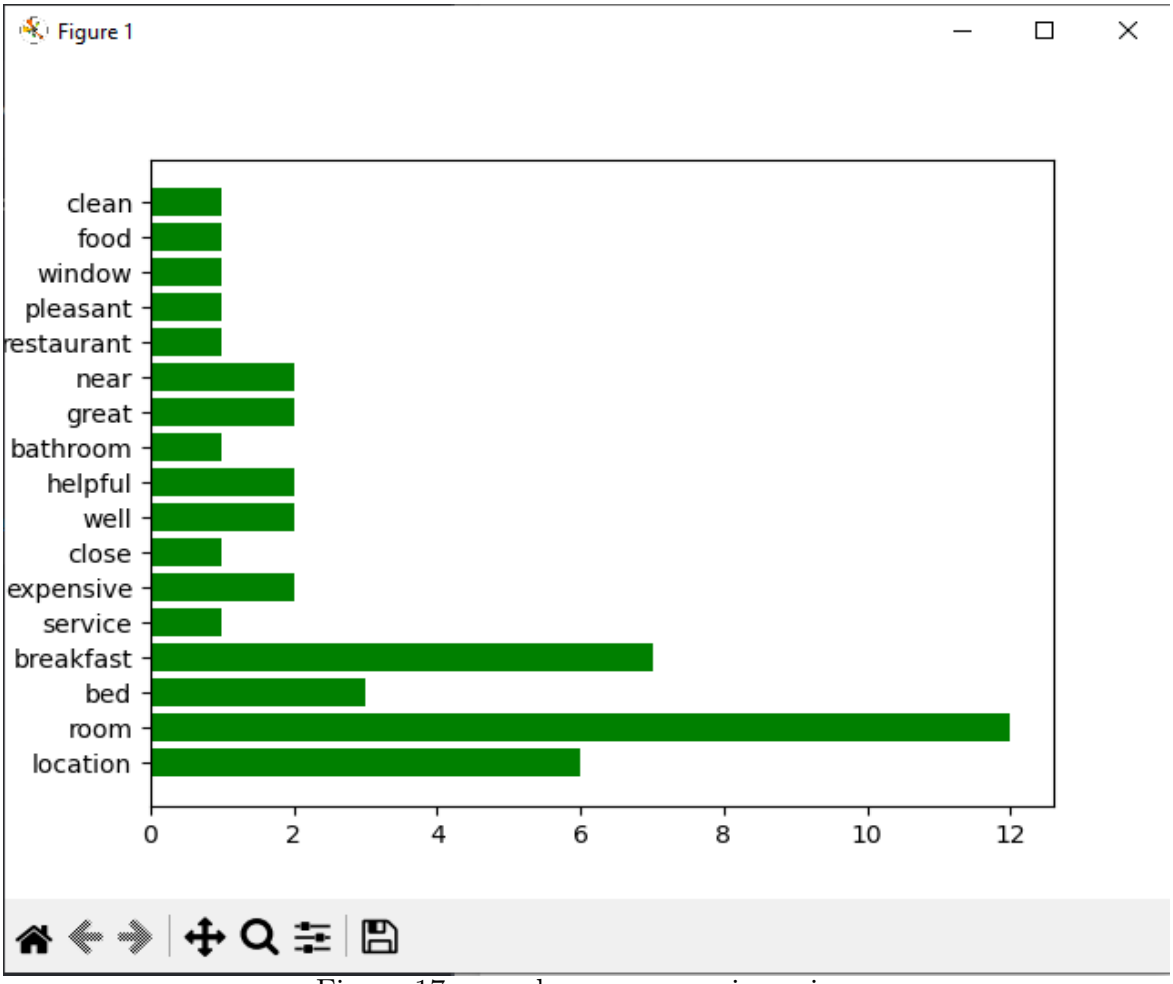

Figure 17 : words appearance in reviews

In the next picture users is able to see the number a category of interest is mentioned in the reviews. We should now mention that in our code we created specific categories related to hotel reviews, where in its category we have included the corresponding words. Below you can find the specific part of the code.

Cleanliness	clean, clear, well, fresh, bedsheet, towel,dirty, mess, smell, badly, iron, cleanness
Facilities	climate, decorated, spa,restaurant, bar, pool, Jacuzzi, Laundry, bathroom, food, breakfast,room,fridge,garden,window,aircondition,aircondition, shower, facecloths, decoration, parking
Staff	service, polite,rude, helpful, pleasant
Location	location, quiet, close ,near, far, distance, transportation,sound, center, long, away, unsecure, dodgy, unsafe
Comfort	bed, security, safety,chill
Value	noise, great, relaxing, expensive, beautiful, horrible, superb, costly

Table 5 : Predefined categories and words

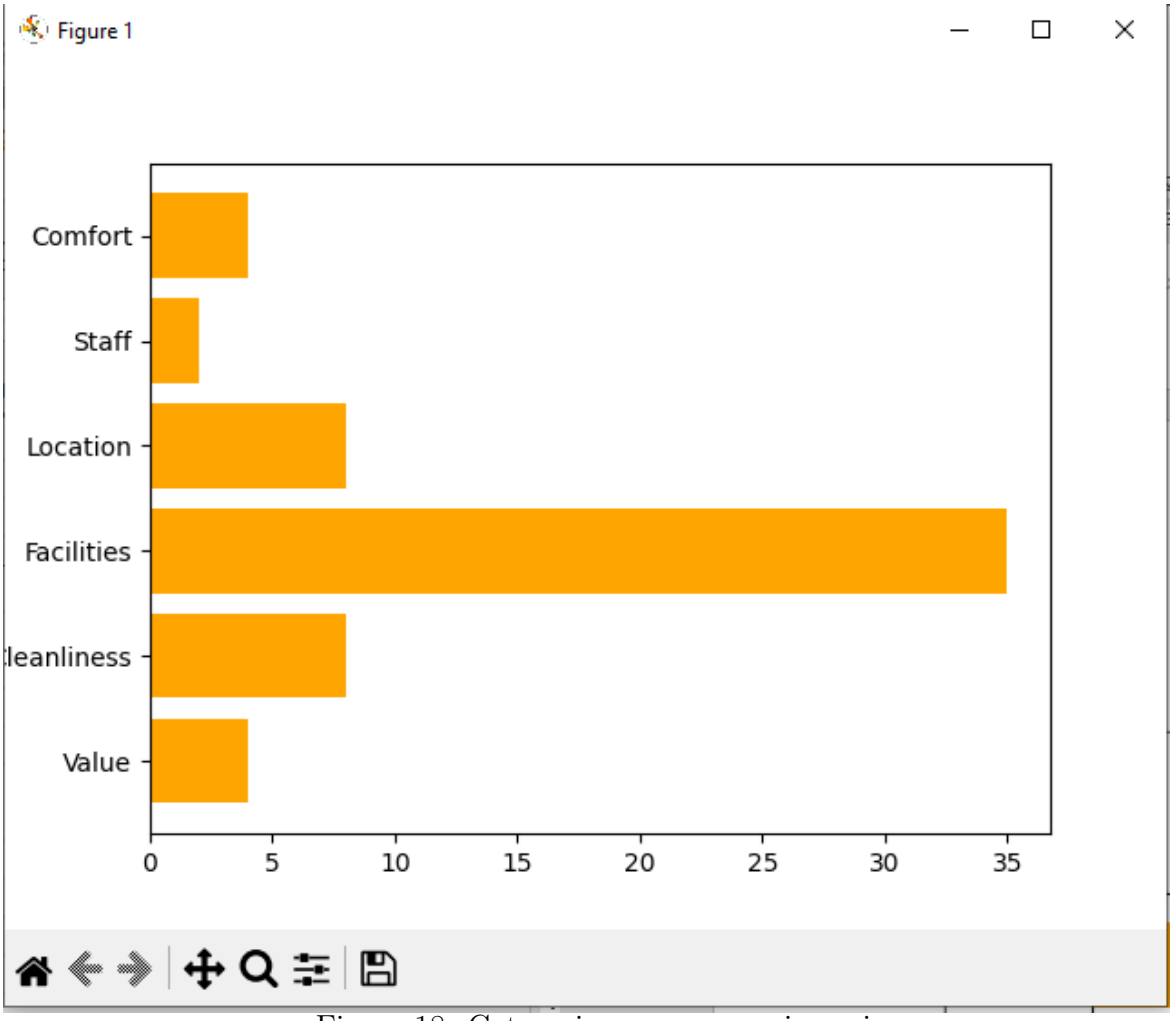

Figure 18: Categories appearance in reviews

Afterwards using VADER[6] sentiment analysis users are able to finger out how many times word of interest appears in review marked as positive , negative or neutral. By clicking in each word tag, the whole reviews in which the specific word appears can be shown in a pop up window. Tags are colored as positive – green , neutral – orange and negative – red in order for user to easily distinguish them.

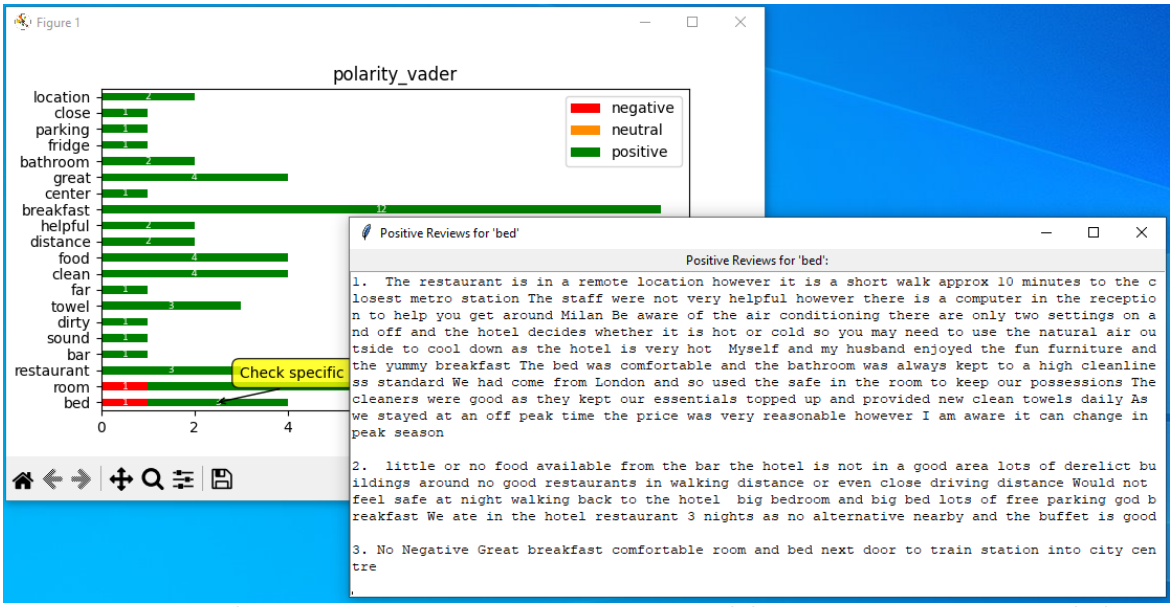

Figure 19 : words appearance in reviews categorized by tags. In case user clicks on a word’s tag , the corresponding reviews appeared

The last graph which is shown in ‘review graphs’ selection is word cloud contains all the word of interest found in the reviews , colored with the tag’s color mainly included.

Figure 20 : Word Cloud from selected hotel reviews

4.5.2 Check reviews and tags

Another option user has, is ‘check reviews and tags’ When user selects this option a new graphs appears that show the relationship between hotel, reviews and tags. On order to achieve this we use a neo4j graph database which we filled with the hotel selected by the user, its reviews and the relevant tag of each review. The nodes of the reviews are colored based on the tag color and when user clicks on a review a new pop up window appears that contains the whole review. The printout can be found in picture x.x

Figure 21 : User Interface main page , check reviews and tags

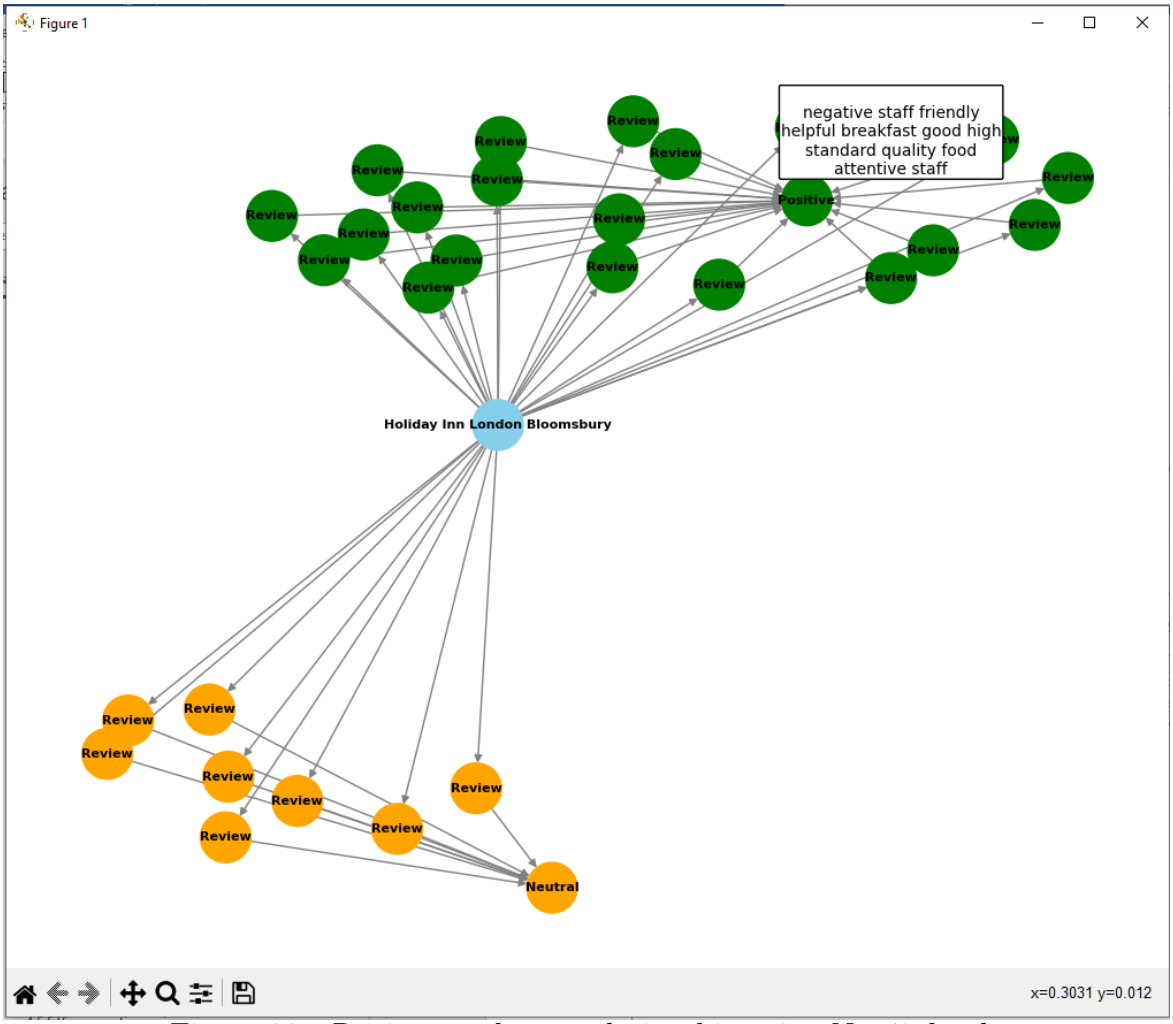

Figure 22 : Reviews and tags relationship using Neo4j database

We selected neo4j database as it is a graph database that uses graph structures to store and query data and python has a Neo4j library provides a convenient interface for interacting with Neo4j databases from Python code. It is designed to efficiently manage and traverse relationships between entities, making it particularly well-suited for use cases involving complex and interconnected data. Data in Neo4j is stored as nodes, which represent entities, and relationships, which represent connections between nodes. This graph-based model allows for flexible and expressive data representation. Neo4j includes a library of graph algorithms

that can be used to analyze and extract insights from graph data. These algorithms cover a wide range of use cases, including path finding, community detection, and centrality analysis.

Neo4j uses Cypher, a declarative query language, to interact with the database. Cypher is designed for readability and expressiveness, making it easy to write and understand complex queries.

4.5.3 View all reviews

On more user selection might be ‘view all reviews’ which will show all the reviews of the specific hotel with no further analyses or cleanness. There are the raw data of the available already posted in booking.com reviews.

Figure 23 : User Interface main page , view all reviews

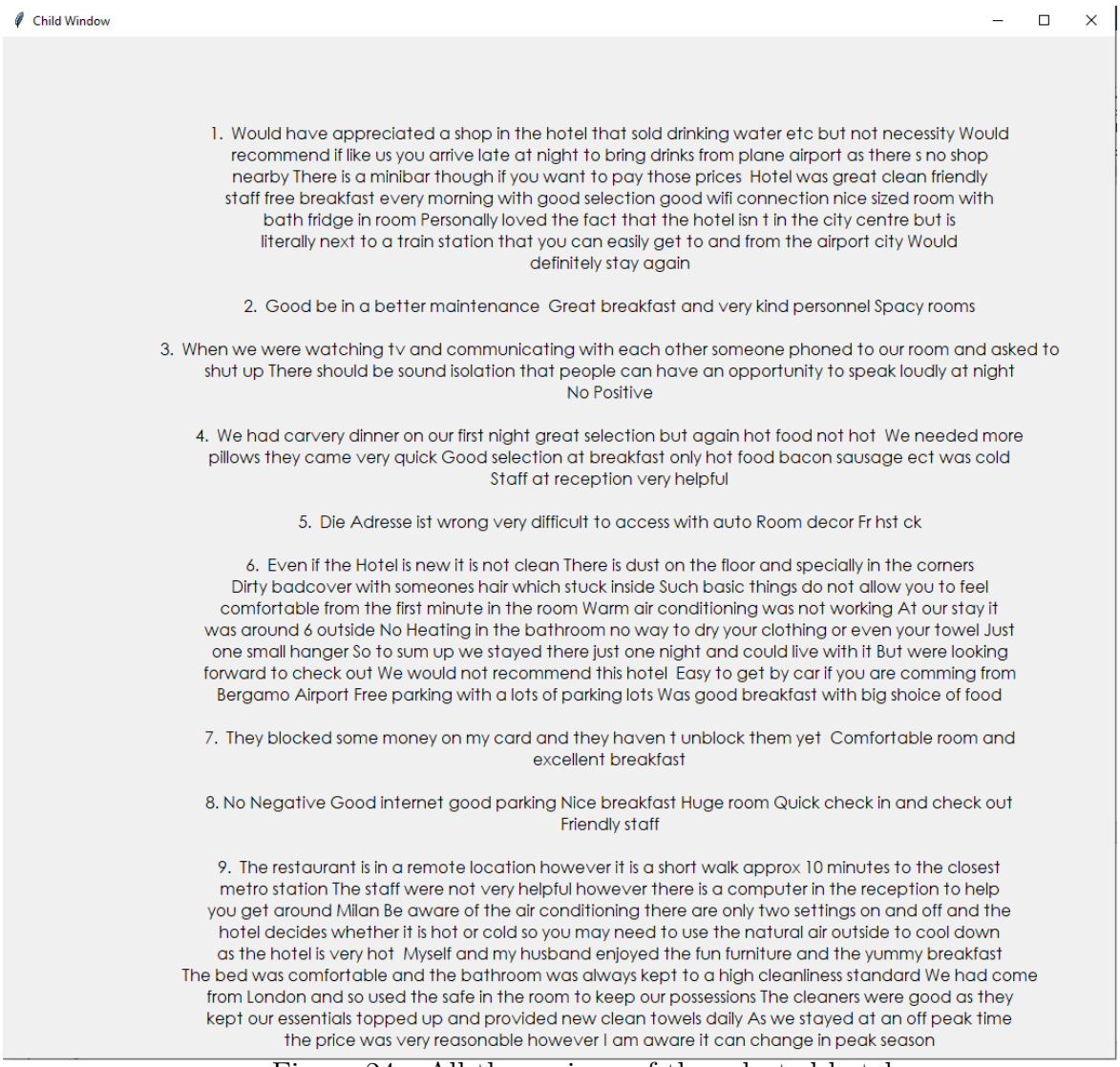

Figure 24 : All the reviews of the selected hotel

4.5.4 View positive reviews

This selection allows users to take a lot only in positive reviews tagged by VADER[6] method. User can check whether the reviews are similar and the most positive benefits of the hotel.

Figure 25 : User Interface main page , view positive reviews

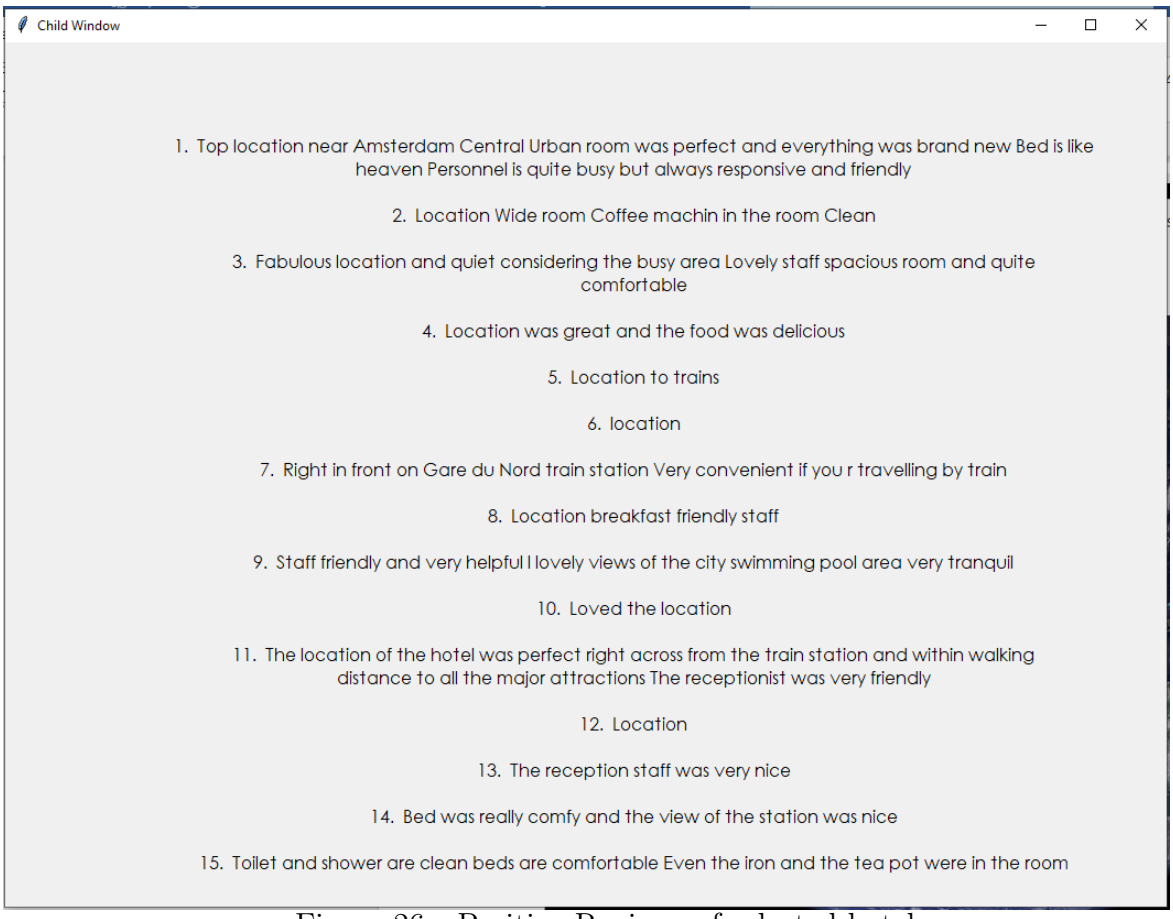

Figure 26 : Positive Reviews of selected hotel

4.5.5 View negative reviews

Similarly to previous option with 'view negative reviews' user is able to check only the available negative reviews of the selected hotel. Obviously user can check the disadvantages of the hotel and decide how important these are for him.

Figure 27 : User Interface main page , view negative reviews

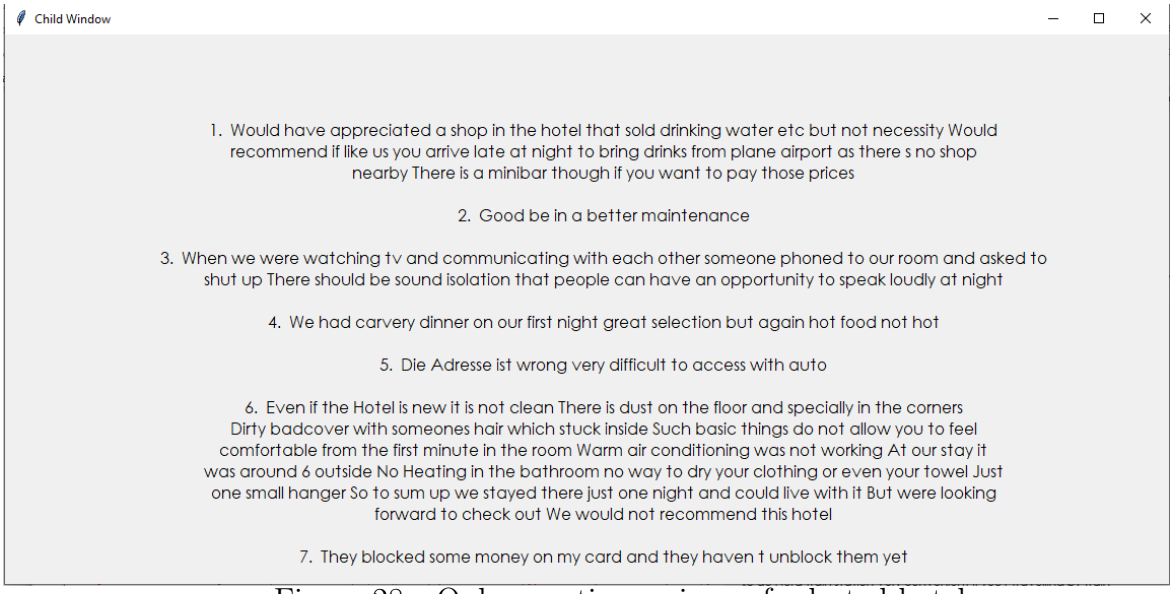

Figure 28 : Only negative reviews of selected hotel

4.5.6 View sum up of reviews

The last option that is available to users is ‘view sum up of the reviews’ which returns a pop up windows with the summary of all the reviews of the selected by the user hotel.

Figure 29 : User Interface main page , view sum up of the reviews

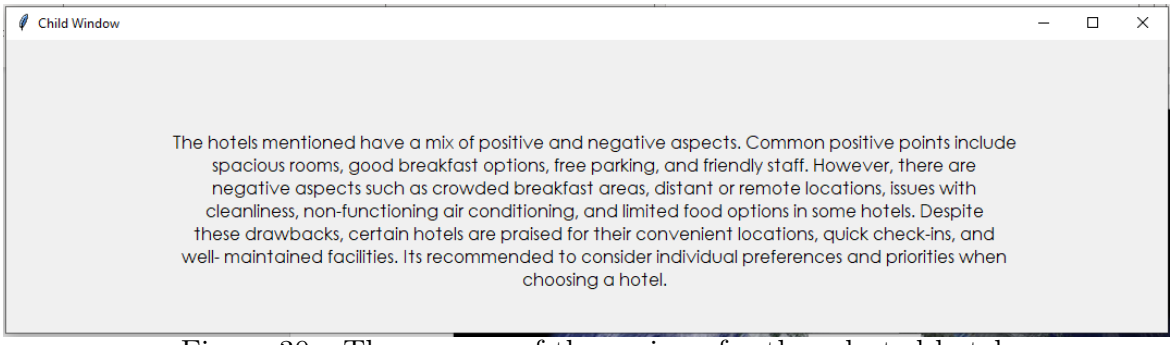

Figure 30 : The sum up of the reviews for the selected hotel

In order to achieve this we deploy an implementation that consumes python openai library. OpenAI Python library provides access to various OpenAI models and APIs, allowing integration to cutting-edge AI capabilities into applications. OpenAI offers a range of models and APIs for natural language processing, text generation, language translation, image recognition, and more.

Specifically in our case we decide to use davinci engine refers to the GPT-3 model developed by OpenAI. "davinci" is a widely-used and powerful model, but there are other models available with different strengths and weaknesses.

5 Conclusion

This thesis has delved into the realm of sentiment analysis, encompassed various facets from understanding sentiment analysis models to examining similarity measures and pre-trained models. The final target was the implementation of an intuitive user interface which provides valuable insights based on existing reviews sentiment analysis for the product or service of interest.

In the investigation of sentiment analysis models, we experiment several kinds of models such as rule-based, machine learning, and hybrid. Additionally, we emphasized the significance of data collection, generative AI, reasoning engines, and explainable AI in enhancing analysis outcomes.

Furthermore, we undertook a comprehensive examination of similarity measures and pre-trained models, focusing on their roles in sentiment analysis process. Specifically, we

examined similarity measures such as WuPalmer[8], Path[3], Resnik (RES)[5], and Leacock-Chodorow-LCH[4], clarifying their applications and implications in sentiment analysis. Moreover, we examine pre-trained models like BERT[7] and VADER[6], so as to highlight their capabilities in sentiment analysis.

The research methodology and experimental setup provided a structured approach to validate the efficiency of our proposed integration. Through meticulous data collection, feature extraction, and model selection processes, we tried to ensure the reliability and robustness of our findings. Additionally, the implementation of the user interface, equipped with features like view graphs, check reviews and tags, and summary of reviews, facilitated seamless interaction and comprehension of sentiment analysis outcomes for end-users.

5.1 System Evaluation

The evaluation of our integrated system revealed promising results in terms of accuracy, efficiency, and user satisfaction. Through testing and analysis, we observed commendable performance metrics, indicating the effectiveness of our approach in capturing and interpreting sentiment dynamics accurately. There are of course pain points that need further analysis such as

- the computer resources, using personal computer we were not able to process huge amount of data. This has as a result, the outcomes to be not so accurate as could be. There were some cases where the computer resources lead to system performance degradation. Users should wait more than average time for the execution of the requested action.

- one of the problems that sentiment analysis has to deal with is the clarification of neutral results, regardless the method used. While it is much easier to clarify and categorize a review to positive or negative it is extremely difficult to do the same for neutral. The test data length along with words used in predefined categories and thresholds affect the number of reviews that system can mark as neutral.

The above issues must be addressed and tried to be eliminated as parts of improvement procedure.

5.2 System Improvements

While our system demonstrates considerable efficacy, there exist avenues for enhancement and refinement. Future iterations could focus on enhancing model interpretability and adapt-

ability, thereby augmenting the explainability and generalization capabilities of the system. It is also important to start with the already noticed defects. We must use more efficient hardware so as to be able to process adequate value of test data, so as to provide users with accurate results in an acceptable time.

We could also implement the online interface towards bookin.com in order to get the latest reviews of the hotels of interest without needing to periodically update the latest csv file. Additionally, leveraging advanced techniques such as active learning and domain adaptation could further enhance the system's performance in diverse contexts. Moreover, continuous user feedback and iterative design processes will be instrumental in refining the user interface to cater to evolving user needs and preferences.

In conclusion, using sentiment analysis tools (in our case VADER[6]) and developing a user interface we perform a significant step towards enhancing sentiment analysis methodologies and applications. Through ongoing evaluation and improvement efforts our goal is to strengthen and enhance the capabilities of our system and improve user experiences.

6 References

[1] Cours de Linguistique Général , 1916 , Albert Sechehaye and Charles Bally

[2] Verb semantics and lexical selection , 1994, Z. Wu and M. Palmer

[3] Path similarity Analysis: A Method for Quantifying Macromolecular Pathways, 2015, Sean L. Seyler, Avishek Kumar,M. F. Thorpe and Oliver Beckstein

[4] Combining local context and WordNet similarity for word sense identification, 1998, Claudia Leacock Martin Chodorow

[5] Using Information Content to Evaluate Semantic Similarity in a Taxonomy, 1995, Philip Resnik.

[6] VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text , 2014, C.J. Hutto Eric Gilbert

[7] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018, Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton

[8] The Relationship Between Precision-Recall and ROC Curves, 2006, Davis Goadrich

[9] The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms , 1997, Andrew P. Bradley

[10] Training language models to follow instructions with human feedback,2022, Ouyang, Long, Wu, Jeff, Jiang, Xu, Almeida, Diogo

[11] Towards Reasoning in Large Language Models: A Survey,2021, Jie Huang KevinChen-Chuan Chang

[12] Sentiment Analysis of Customer Reviews of Food Delivery Services Using Deep Learning and Explainable Artificial Intelligence: Systematic Review, 2022 , Anirban Adak

[13]. Interpretable AI: Building Explainable Machine Learning Systems, 2022, Ajay Thampi

[14]. Machine Learning for Beginners: Build and Deploy Machine Learning Systems Using Python, 2nd Edition, 2023, Dr. Harsh Bhasin

[15]. Applied Deep Learning: Design and Implement Your Own Neural Networks to Solve Real-World Problems , 2023 , Dr. Rajkumar Tekchandani, Dr. Neeraj Kumar

[16]. Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data, 2016, Dipanjan Sarkar

[17]. Handbook of Research on Opinion Mining and Text Analytics on Literary Works and Social Media, 2022, Pantea Keikhosrokiani, Moussa Pourya Asl

[18] Natural Language Processing and Machine Learning for Developers, 2021, Oswald Campesato

[19] Sentiment Analysis in Social Networks , 2016, Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, Bing Liu

[20] Implement NLP[1] Use-Cases Using BERT: Explore the Implementation of NLP[1] Tasks Using the Deep Learning Framework and Python , 2021, Amandeep

[21] Natural Language Processing Fundamentals for Developers , 2021, Oswald Campesato

7 Appendix A: “Part of the Code”

Below you find parts of the code while the whole py files can be found in https://github.com/ChristinaKatsir

I. Import and clean up data

```
reviews_df = pd.read_csv("C:/Users/katsi/Desktop/paradotea_local/Hotel_Review
reviews_df["review_total"]=reviews_df["Negative_Review"] + reviews_df["Positiv
print(reviews_df)
reviews_df = reviews_df.sample(frac = 0.008, replace = False, random_state=42
reviews_df["review_total"] = reviews_df["review_total"].fillna('')

def get_wordnet_pos(pos_tag):
    if pos_tag.startswith('J'):
        return wordnet.ADJ
    elif pos_tag.startswith('V'):
        return wordnet.VERB
    elif pos_tag.startswith('N'):
        return wordnet.NOUN
    elif pos_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN

def clean_text(text):
    # lower whole text
    text = text.lower()
    # tokenize text and remove puncutation....word.strip removes all the punc
    text = [word.strip(string.punctuation) for word in text.split(" ")]
    # remove words that contain numbers
    text = [word for word in text if not any(c.isdigit() for c in word)]
    # remove stop words
    stop_english = stopwords.words('english')
    stop_greek = stopwords.words('greek')
    text = [x for x in text if x not in stop_english]
```

```
    text = [x for x in text if x not in stop_greek]
    # remove empty tokens
    text = [t for t in text if len(t) > 0]
    # pos tag text ... find the grammar meaning of each word
    pos_tags = pos_tag(text)
    # lemmatize text ...meaning that words are replaced with more common with
    text = [WordNetLemmatizer().lemmatize(t[0], get_wordnet_pos(t[1])) for t
    #print(text)
    # remove words with only one letter
    text = [t for t in text if len(t) > 1]
    # join all
    text = " ".join(text)
    return(text)

reviews_df["positive_clean"] = reviews_df["Positive_Review"].apply(lambda x:
reviews_df["negative_clean"] = reviews_df["Negative_Review"].apply(lambda x:
reviews_df["review_total_clean"]= reviews_df["review_total"].apply(lambda x:
```

II. Create Categories of Interest

```
categories = {
    'Cleanliness': ['clean', 'clear', 'well', 'fresh', 'bedsheet', 'towel','d
    'Facilities': ['climate', 'decorated', 'spa','restaurant', 'bar', 'pool',
    'Staff': ['service', 'polite','rude', 'helpful', 'pleasant'],
    'Location': ['location', 'quiet', 'close' ,'near', 'far', 'distance', 'tr
    'Comfort': ['bed', 'security', 'safety','chill'],
    'Value':['noise', 'great', 'relaxing', 'expensive', 'beautiful', 'horribl
}


terms_df= pd.DataFrame(
             [(k, val) for k, vals in categories.items() for val in vals],
             columns=['category', 'term']
            )
```

II. Create the connection Towards neo4j DataBase

```
from neo4j import GraphDatabase
    driver = GraphDatabase.driver("bolt://localhost:7687", auth = basic_auth(
    session = driver.session()
```

III. Use VADER in order to find the reviews tag

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer

sid = SentimentIntensityAnalyzer()
#create a new column (sentiments that keeps data created by polarity feature,
reviews_df_temp["sentiments"] = reviews_df_temp["review_total_clean"].apply(la
```

IV. Use Tk library for creating User Interface Functions

```
root = tk.Tk()
root.title("tk")
root.geometry("900x900")
# Add image file
from PIL import ImageTk, Image
bg = ImageTk.PhotoImage(Image.open("C:/Users/katsi/Desktop/paradotea_local/pe

# Show image using label
label1 = Label( root, image = bg)
label1.place(x = 0, y = 0)

label2 = Label( root, text = "Welcome.\nSelect from the dropdown menu below t
label2.pack(pady = 50)

# Entry for searching
label3 = Label(root, text="Search Hotel:")
label3.pack()

entry = Entry(root)
entry.pack()
```

```
entry.bind('<KeyRelease>', Scankey)
listbox = Listbox(root)
listbox.pack()
Update(values1)

# Create Frame
frame = Frame(root)
frame.pack(pady = 20)

# Create label
#selected1 = tk.StringVar()
#selected1.set("Select Hotel")

#options1 = tk.OptionMenu(root, selected1, *values1)
#options1.pack()

selected2 = tk.StringVar()
selected2.set("Select Action")

options2 = tk.OptionMenu(root, selected2, *values2)
options2.pack()

button5 = tk.Button(root, text='help', command=on_click5)
button5.place(x=450,y=850)

# Button for searching and selecting
#button9 = tk.Button(root, text="Search and Select", command=search_and_selec
#button9.place(x=450,y=500)

button8 = tk.Button(root, text='selected action', command=on_click8)
button8.place(x=405,y=550)


root.mainloop()
```

Author's Statement:

I hereby expressly declare that, according to the article 8 of Law 1559/1986, this dissertation is solely the product of my personal work, does not infringe any intellectual property, personality and personal data rights of third parties, does not contain works/contributions from third parties for which the permission of the authors/beneficiaries is required, is not the product of partial or total plagiarism, and that the sources used are limited to the literature references alone and meet the rules of scientific citations.