

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Djilali BOUNAAMA Khemis Miliana



Faculty of Science and Technology
Department of Mathematics and Computer Science

Thesis presented to obtain

a Master degree in

« Computer science »

Specialty: Software Engineering and Distributed Systems

Title

**Explaining and visualizing deep learning models for
sentiment analysis**

Presented by :

- BENAMEUR Mohamed
- KHLIDJ Mohammed

Supervised by:

Mr. AZZOUZA Nouredine

Academic year 2022/2023

Dédicaces

I dedicate this work to:

*The two most dear beings in the world, the candles who have always
guided me on the right path:*

*My mother, the light of my life, who did every thing for my success
and my happiness.*

*My father, to him, we must offer all the respect and love for his
support and tenderness.*

To my brother for their help and and encouragement.

To all family members and friends.

Benameur

I dedicate this modest work:

To all my family members for their continued support and i wish them good health and long life To all my friends, especially my friend Mohamed Benameur who has been my companion for several years.

To all my teachers who did their best to give us as much information as possible about our study.

Finally, to all those who have gained directly or indirectly by accomplishing this work.

Khlij

Acknowledgement

First of all, we would like to thank Almighty Allah for giving us the strength, knowledge, ability and patience to achieve this modest work, and without his blessings this achievement would not have been possible.

We would like to send our sincere and warm thanks to our research promoter **Mr.” AZZOUZA Noureddine”**, of who we were lucky to have him as a supervisor and who kindly entrusted us with this work, rich in experience and guide us in each step of the process.

He gave us all the freedom to pursue our research. We express our thanks to him here for his constant orientations of our research, as well as for his invaluable advice, his availability and his extreme kindness.

Our most sincere thanks, as well as our gratitude's go to our parents as well as our brothers and sisters, who have accompanied and supported us throughout our journey.

We would like to take this opportunity and send our deep thanks to all the teaching, administrative and educational staff of the mathematics and computer science department at the University of Djilali BOUNAAMA Khemis Miliana.

Sommaire

Abreviations list.....	VII
List of Tables.....	VIII
List of Figures	VIII
Introduction	11
I. State of the art	- 3 -
I.1. Sentiment analysis	- 3 -
I.1.1. introduced	- 3 -
I.1.2. The application areas of sentiment analysis	- 3 -
I.1.3. Approaches deep learning models sentiment analysis.....	- 5 -
I.1.4. Data Collection and Preprocessing.....	- 8 -
I.1.4.1. Introduction.....	- 8 -
I.1.4.2. Importance of Data Collection and Preprocessing in Sentiment Analysis	- 8 -
I.1.4.3. Data Collection	- 9 -
I.1.4.3.1. Challenges of collecting data for sentiment analysis.....	- 9 -
I.1.4.3.2. Data collection from social media platforms like Twitter	- 9 -
I.1.4.4. Data Preprocessing.....	- 10 -
I.1.4.4.1. Steps in Data Preprocessing	- 11 -
I.2. Explaining deep learning models	- 12 -
I.2.1. Introduction.....	- 12 -
I.2.2. The applications of explaining deep learning models.....	- 13 -
I.2.3. Visualize Deep Learning Models	- 14 -
I.2.4. Tools of explaining deep learning models.....	- 16 -
Interpretable Model-Agnostic Explanations (LIME).....	- 16 -
1.1.1 SHAP (SHapley Additive exPlanations).....	- 17 -
Conclusion	- 20 -
II. Analyse et conception	- 22 -
II.1. Introduction	- 22 -
II.2. Definition.....	- 22 -
II.3. Architecture du système	- 22 -
II.4. Steps of the process	- 23 -
II.5. Model used for classification	- 24 -
II.5.1. Model01	- 26 -

II.5.2. Model02	- 26 -
II.6. explaining tools used.....	- 27 -
III. Implementation and discussion	- 32 -
III.1. Introduction	- 32 -
III.2. The tools and libraries used	- 32 -
III.2.1. Langage	- 32 -
III.2.2. Tools	- 32 -
III.2.3. Libraries	- 32 -
III.3. Datasat	- 34 -
III.4. Results :	- 35 -
III.4.1. Evaluation metrics.....	- 35 -
III.4.2. Implementation.....	- 36 -
Model 01 :	- 36 -
Results :	- 36 -
Model 02 :	- 37 -
Results :	- 38 -
III.4.3. Discussions	- 39 -
Model 01 :	- 39 -
Model 02 :	- 41 -
Model 02 :RoBERTa.....	- 50 -
conclusion	- 44 -

Abbreviations list

CNN: Convolutional Neural Networks

ReLU: Rectified Linear Unit

LSTM: Long Short-Term Memory

RNN: Recurrent Neural Network

SVM: Support Vector Machine

NN: Neural Network

API: *Application Programming Interface*

RAM: Random Access Memory

LIME: Interpretable Model-Agnostic Explanations

SHAP: SHapley Additive exPlanations

MPL: Pre-trained Language Model.

XAI: explainable Artificial Intelligence

Grad-CAM: Gradient-weighted Class Activation Mapping

T-SNE Visualization: T-distributed Stochastic Neighbor Embedding

List of Tables

Tableau III-1 Dataset description SemVal2017	- 35 -
Tableau III-2 Results of model BERT	- 36 -
Tableau III-3 Results of model RoBERTa	- 38 -

List of Figures

Figure I-1 : Classification of feelings of some sentences[17]	- 3 -
Figure I-2: Architecture of a Convolutional Neural Network (CNN)[19]	- 5 -
Figure I-3: Long Short-Term Memory (LSTM) Network for Sentiment Analysis[21]	- 6 -
Figure I-4: Architecture of BERT Model for Sentiment Analysis[22]	- 7 -
Figure I-5: Overall pre-training and fine-tuning procedures for BERT[26]	- 7 -
Figure I-6 Data Collection and Preprocessing[27]	- 8 -
Figure I-7 Data Preprocessing[28]	- 10 -
Figure I-8 : Stages of Data Preprocessing [29]	- 11 -
Figure I-9: The XAI Concept[30]	- 13 -
Figure I-10: Explaining the prediction of a random forest classifier on a Yelp review	- 17 -
Figure I-11 Shap-explained model deep learning models[39]	- 19 -
Figure I-12 Layer-wise Relevance Propagation (LRP) applied onto various neural[40]	- 19 -
Figure II-1 Representation of the system architecture[41]	- 23 -
Figure II-2 How Transformers work in deep learning and NLP[43]	- 25 -
Figure II-3 Fine - tuning - based classification of a PLM[44]	- 25 -
Figure II-4 Configuration of model 01	- 26 -
Figure II-5 Configuration of model 02	- 26 -
Figure III-1 Accuracy and Error for the Bert epoch=4	- 39 -
Figure III-2 Accuracy and Error for the RoBERTa epoch=4	- 41 -
Figure III-3 Part of the dataset testing	- 43 -

الملخص

تحليل المشاعر في التعلم العميق هو مجال يهدف لتقييم الحالة العاطفية لنصوص معينة. يشمل استخدام أساليب متنوعة لمعالجة المحتوى النصي وتضمين العبارات، ويستعين بشبكات عصبية مثل RNNs و CNNs ونماذج تحويل مثل BERT و RoBERTa. تحتاج النماذج لتدريب على مجموعات كبيرة من النصوص المعروفة بالعاطفة (إيجابية وسلبية ومحايدة). يُعد تعقيد اللغة الطبيعية تحديًا، ولذلك يتم استخدام تضمين الكلمات لالتقاط الروابط الدلالية بينها. تستخدم هندسة الشبكة العصبية لكشف الأنماط المحلية والتبعيات ذات المدى الطويل في النصوص. يستخدم تصور الانتباه وخرائط الأهمية لتوضيح العمليات الداخلية للنماذج وتحسين قابلية تفسيرها.

Résumé

L'analyse des sentiments avec des modèles d'apprentissage en profondeur évalue les émotions d'un texte en utilisant des prétraitements, des plongements de phrases et des réseaux neuronaux comme CNN et RNN. Ils sont formés sur de vastes ensembles de données étiquetées pour chaque émotion. Les plongements de mots sont utilisés pour saisir les liens sémantiques dans le texte, tandis que les CNN détectent les motifs locaux et les RNN identifient les dépendances à long terme. Les techniques de visualisation telles que la visualisation de l'attention et les cartes de saillance aident à comprendre le fonctionnement interne de ces modèles. En résumé, les modèles d'apprentissage en profondeur facilitent l'identification automatisée des émotions dans un texte et permettent de comprendre la complexité du langage naturel.

Abstract

Sentiment analysis is a vital application of deep learning models, aiming to assess emotional tones in texts. These models combine content preprocessing, phrase embeddings, and neural network topologies like CNNs and RNNs, trained on labeled datasets with recognized emotions (positive, negative, or neutral). To tackle the complexity of natural language, word embeddings are utilized to capture semantic links between words. CNNs excel in detecting local patterns, while RNNs are better at identifying long-term dependencies in text. Visualizations like attention visualization and saliency maps offer insights into model workings and enhance interpretability. Overall, deep learning models for sentiment analysis automate emotional tone identification, capturing language complexity effectively.

Keywords: sentiment analysis, interpretability, visualization, NLP, deep learning, Neural networks, analyze feelings.

Introduction

Introduction

Sentiment analysis, also known as opinion mining, is an intriguing field of study that delves into the intricate nuances of human expression[1]. It aims to decipher the sentiments, emotions, attitudes, and appraisals conveyed through written text. By analyzing and understanding the overall sentiment be it positive, negative, or neutral sentiment analysis provides valuable insights into people's opinions and perceptions.

The origins of sentiment analysis can be traced back to [2], who first introduced the phrase, while the term "opinion mining" was coined by[3].Nevertheless, the exploration of sentiment and opinion predates these seminal works, with earlier research conducted by [1], [4]–[9]. Furthermore, the realm of sentiment analysis encompasses broader domains, such as metaphor interpretation, sentiment adjective extraction, affective computing, subjectivity analysis, viewpoints, and affects, as explored by[10]–[15]. Notably, an early patent on text classification by[16] even considered sentiment, humor, and other concepts as possible class labels.

Initially, sentiment analysis primarily focused on textual data, propelling it into the forefront of natural language processing (NLP) research. However, in recent years, sentiment analysis has ventured beyond the realm of written text to encompass multimedia content such as photos, videos, and audio recordings. This expansion allows for a more comprehensive analysis of people's sentiments and opinions. The advent of social media platforms like Facebook and Twitter in the mid-2000s ushered in a new era for sentiment analysis, fueled by the vast amounts of data these platforms generated[1].

Deep learning techniques, particularly neural networks, have emerged as powerful tools in sentiment analysis, revolutionizing its landscape. The integration of these techniques has yielded remarkable improvements in accuracy and performance, effectively pushing the boundaries of sentiment analysis systems. However, sentiment analysis remains a dynamic field, continuously evolving as researchers explore novel techniques and methodologies to unravel the complex web of sentiment in text.

Explaining and visualizing deep learning models is an essential aspect of modern artificial intelligence research. Deep learning models, inspired by the human brain's neural networks, have achieved remarkable success in various tasks such as image recognition, natural language processing, and speech synthesis. While these models can deliver highly accurate predictions, their inherent complexity often makes them appear as "black boxes," lacking transparency in how they

arrive at their decisions. To address this challenge, researchers have been actively developing techniques to explain and visualize the inner workings of deep learning models, providing insights into their decision-making processes and enhancing their interpretability. By shedding light on the learned representations and decision paths, these efforts not only build trust in AI systems but also pave the way for more responsible and effective use of deep learning in diverse domains.

The applications of sentiment analysis are wide-ranging and impactful. For instance, in the realm of customer feedback analysis, sentiment analysis enables businesses to harness the knowledge and classification of customer opinions, thereby facilitating service improvements. Social media monitoring stands as another vital application, allowing companies to navigate the vast expanse of user-generated content including posts, comments, and reviews by utilizing sentiment analysis to gauge public perception of their brand, product, or service. Moreover, sentiment analysis serves as a powerful tool in political analysis, providing insights into public opinion and voting trends through the analysis of political speeches, debates, and social

media activity. In the realm of market research, sentiment analysis empowers organizations to uncover consumer trends and preferences by examining customer comments, reviews, and social media activity[1], [4]–[6], [8], [9].

Undoubtedly, sentiment analysis plays a pivotal role in deciphering the thoughts and emotions conveyed through textual data, offering invaluable insights to businesses, organizations, and researchers. By harnessing the power of sentiment analysis, stakeholders can make informed decisions, enhance customer satisfaction, and identify untapped opportunities for growth. Nevertheless, the remarkable achievements of deep learning models, particularly neural networks, come with a trade-off interpretability. These models, often perceived as enigmatic "black boxes," present challenges in critical domains like law, healthcare, and finance, where understanding the underlying rationale behind model decisions is of utmost importance [1], [5], [9]

My memoir is structured in a logical manner, starting with the introduction and background, followed by the explanation of deep learning models and their applications in sentiment analysis. Then, it delves into the analysis and design of the sentiment analysis system, including the models used and the explanation tools applied. Next, the implementation and results are presented and discussed in detail. Finally, the conclusion summarizes the key findings and provides insights into future research possibilities.

Chapter I

State of the art

I. State of the art

I.1. Sentiment analysis

I.1.1. introduced

Sentiment analysis, also known as opinion mining, is a computational technique used to determine the sentiment or emotion expressed in a piece of text. It involves analyzing and extracting subjective information from textual data to understand the overall sentiment conveyed, whether it is positive, negative, or neutral.

The goal of sentiment analysis is to automate the process of gauging opinions, attitudes, and emotions expressed by individuals or groups towards a particular topic, product, service, or event. This analysis can be applied to various forms of text, such as customer reviews, social media posts, news articles, surveys, and more.

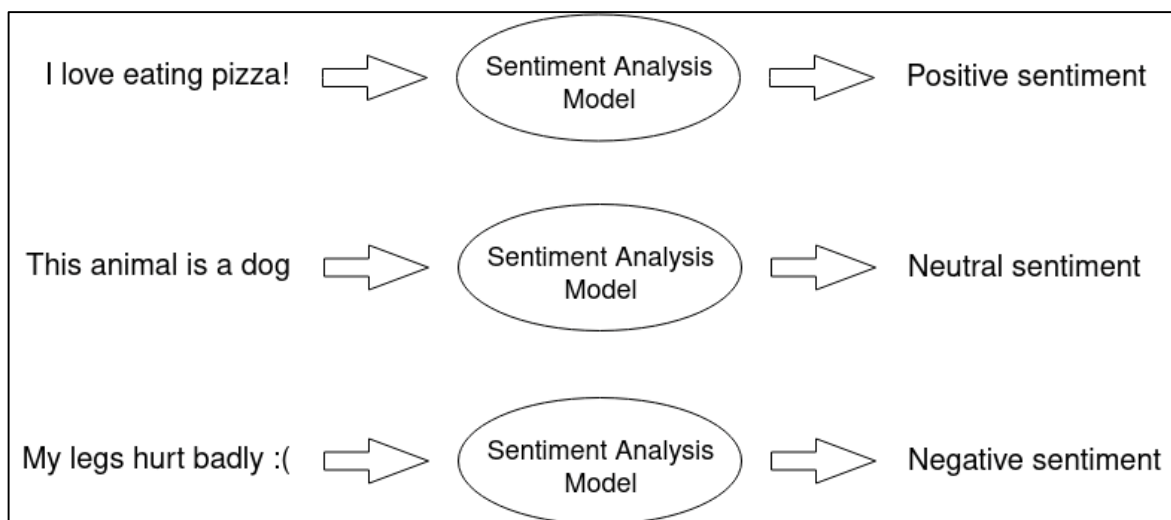


Figure I-1 : Classification of feelings of some sentences[17]

I.1.2. The application areas of sentiment analysis

Sentiment analysis finds applications in a wide range of fields and industries. Here are some key areas where sentiment analysis is commonly used:

Social Media Monitoring: Sentiment analysis helps monitor and analyze public opinion on social media platforms such as Twitter, Facebook, and Instagram. It enables businesses to track brand mentions, assess customer sentiment, and respond to customer feedback in real-time.

Customer Feedback and Reviews: Sentiment analysis is used to analyze customer reviews, feedback surveys, and online ratings. It helps businesses understand customer satisfaction levels, identify areas for improvement, and make data-driven decisions to enhance their products or services.

Brand Reputation Management: Companies use sentiment analysis to monitor their brand reputation by analyzing online conversations, news articles, and blog posts. It helps them identify and address any negative sentiment or potential PR crises, allowing them to take proactive measures to protect their brand image.

Market Research: Sentiment analysis is utilized in market research to gather insights about consumer preferences, behavior, and trends. By analyzing customer sentiments expressed in online discussions and reviews, businesses can understand market sentiment towards specific products, competitors, or emerging trends.

Political Analysis: Sentiment analysis plays a crucial role in political campaigns and public opinion monitoring. It helps political parties and candidates gauge public sentiment towards specific policies, candidates, or events. It can also be used to analyze social media conversations during elections or political debates.

Customer Service and Support: Sentiment analysis is employed in customer service to analyze customer inquiries, emails, and chat logs. It helps identify customer sentiment and detect potential issues or dissatisfied customers, allowing support teams to prioritize and address their concerns more effectively.

Financial Markets: Sentiment analysis is used in the financial industry to analyze news articles, social media posts, and financial reports. It helps traders and investors gauge market sentiment, make informed investment decisions, and identify potential market trends.

Product Feedback and User Experience: Sentiment analysis is used by product teams to analyze user feedback, product reviews, and customer surveys. It provides insights into user sentiment, feature preferences, and areas of improvement, helping companies enhance their products and provide a better user experience.

Healthcare and Public Health: Sentiment analysis is employed in healthcare to analyze patient feedback, social media discussions, and online health forums. It helps identify trends in

patient sentiment, public health concerns, and potential outbreaks by monitoring discussions related to symptoms, treatments, and healthcare providers.

These are just a few examples of how sentiment analysis is utilized across various domains. The ability to understand and analyze sentiment in textual data has proven to be valuable in decision-making, customer engagement, and gaining insights into public opinion.

I.1.3. Approaches deep learning models sentiment analysis

Deep learning models have been increasingly employed for sentiment analysis due to their capacity to understand complicated patterns in text data. Deep learning techniques for sentiment analysis include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models such as BERT.

By applying convolutional filters to the input text, CNNs may learn hierarchical features. [18] suggested a simple and effective CNN architecture for phrase categorization that outperformed the competition on numerous benchmark.

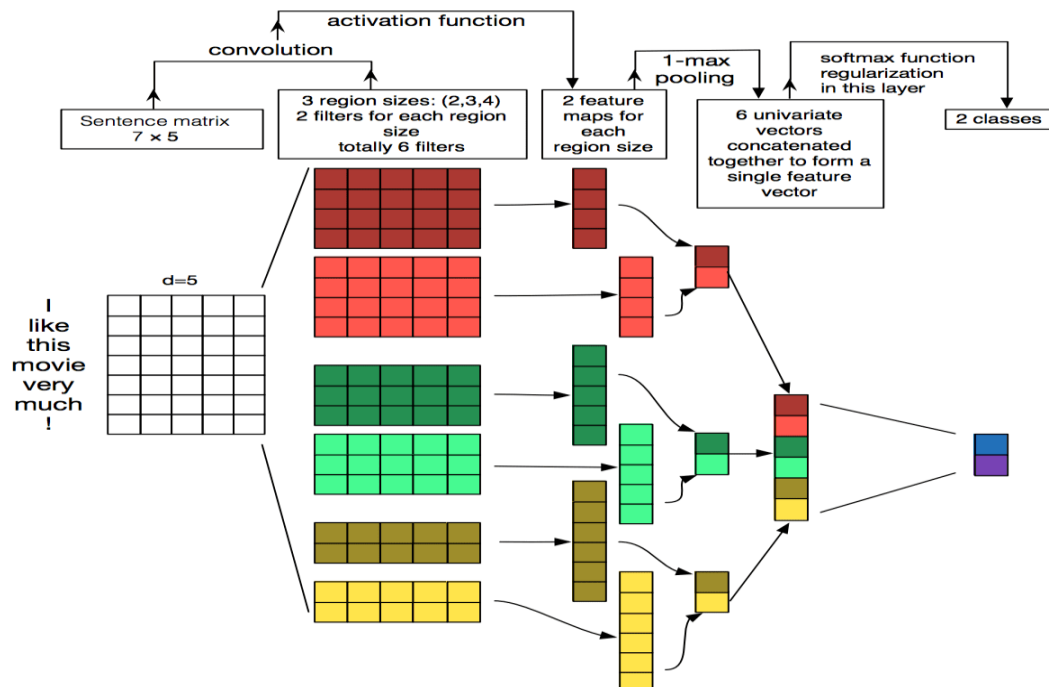


Figure I-2: Architecture of a Convolutional Neural Network (CNN)[19]

datasets. RNNs, on the other hand, can capture the temporal connections between words in a text, making them suited for sentiment analysis jobs where the context of a phrase is important[20]. demonstrated that LSTM networks outperformed established machine learning algorithms such as Support Vector Machines (SVMs) for sentiment analysis.

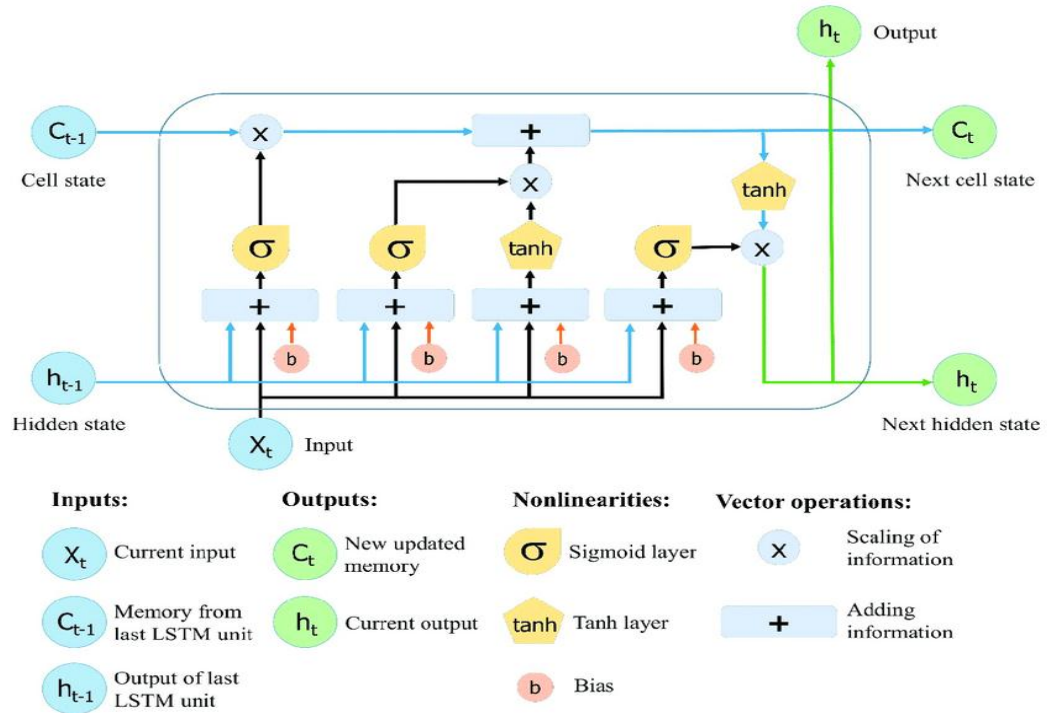


Figure I-3: Long Short-Term Memory (LSTM) Network for Sentiment Analysis[21]

BERT and other transformer-based models have showed promising results on a variety of natural language processing applications, including sentiment analysis. BERT has been pre-trained on vast volumes of text data and has been fine-tuned for certain sentiment analysis

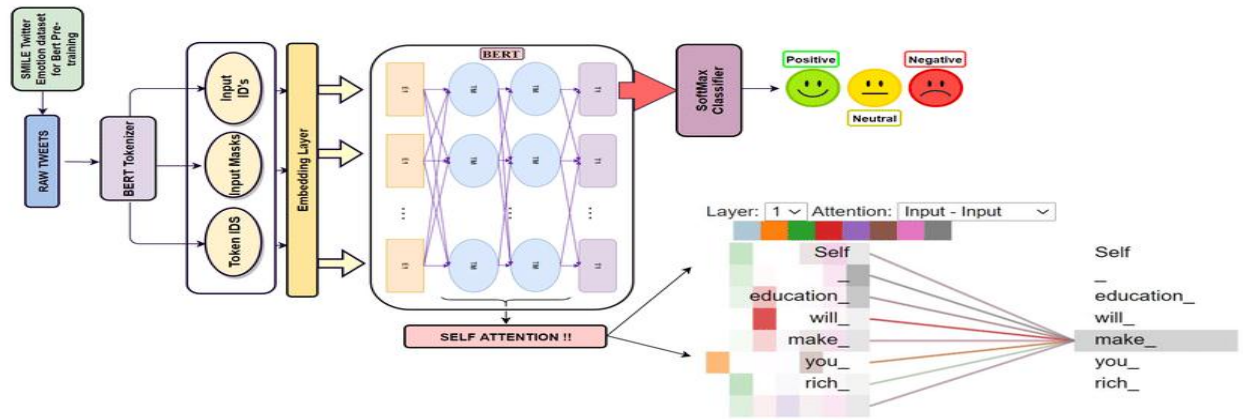


Figure I-4: Architecture of BERT Model for Sentiment Analysis[22]

applications. BERT, suggested by[23], demonstrated state-of-the-art performance on numerous natural language processing tasks, including sentiment analysis. Furthermore, transformer-based models such as XLNet[24] and RoBERTa [25] that improve on the BERT architecture have been presented.

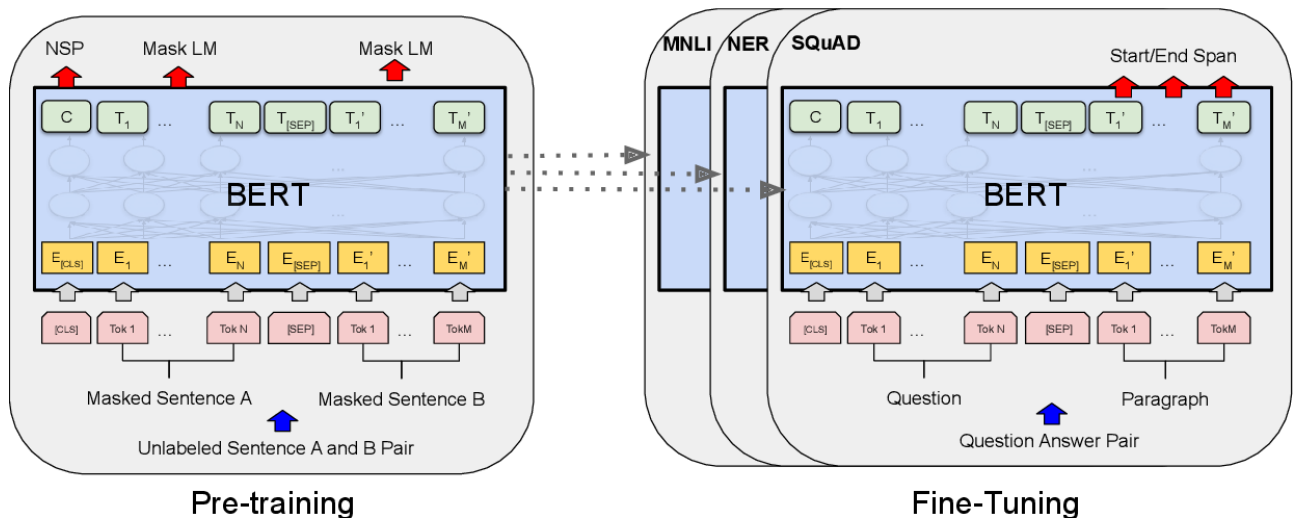


Figure I-5: Overall pre-training and fine-tuning procedures for BERT[26]

I.1.4. Data Collection and Preprocessing

I.1.4.1. Introduction

Data collection and preprocessing are critical components of sentiment analysis, a technique used to identify and extract subjective information from text data. Sentiment analysis is widely used in fields such as marketing, customer service, and political analysis to understand people's opinions, attitudes, and emotions towards a particular product, service, or topic. In this article, we will discuss the importance of data collection and preprocessing in sentiment analysis.

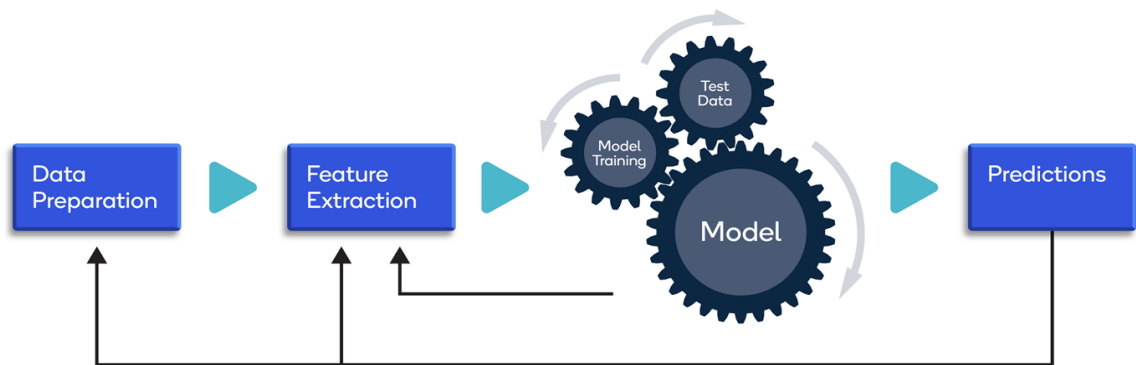


Figure I-6 Data Collection and Preprocessing[27]

I.1.4.2. Importance of Data Collection and Preprocessing in Sentiment Analysis

The accuracy of sentiment analysis largely depends on the quality and quantity of data collected for analysis. Data collection involves identifying sources of data that contain relevant and unbiased information for analysis. For instance, social media platforms like Twitter, Facebook, and Instagram are rich sources of data for sentiment analysis as they provide real-time and diverse opinions from people all around the world. However, collecting data from these platforms can be challenging as they contain vast amounts of data that can be noisy, unstructured, and biased.

Preprocessing, on the other hand, involves cleaning and preparing data for analysis. It involves various techniques such as tokenization, stopword removal, and stemming to convert raw text data into a structured format that can be analyzed by a machine learning model. Data preprocessing helps to improve the accuracy of sentiment analysis by removing irrelevant information, reducing data dimensionality, and standardizing data to a consistent format.

1.1.4.3. Data Collection

Sentiment analysis is a technique used to analyze people's opinions, attitudes, and emotions towards a particular product, service, or topic. To perform sentiment analysis, organizations need to collect data from various sources that contain relevant and unbiased information for analysis. In this chapter, we will discuss the various sources of data that can be used for sentiment analysis.

1.1.4.3.1. Challenges of collecting data for sentiment analysis

Collecting data for sentiment analysis can be challenging as it requires organizations to identify sources of data that contain relevant and unbiased information for analysis. In this article, we will discuss the challenges of collecting data for sentiment analysis.

Challenges of collecting data for sentiment analysis:

Bias: Data collected for sentiment analysis can be biased towards a particular group or demographic, leading to inaccurate results.

Privacy concerns: Collecting data for sentiment analysis can raise privacy concerns as organizations need to ensure that they are not collecting sensitive or personal information without consent.

Sample size: The sample size of data collected for sentiment analysis can impact the accuracy of results. A small sample size may not be representative of the entire population.

1.1.4.3.2. Data collection from social media platforms like Twitter

Social media platforms like Twitter are a rich source of data for sentiment analysis. In this article, we will provide a step-by-step guide on how to collect data from Twitter for sentiment analysis.

Data collection from Twitter for sentiment analysis:

Set up a Twitter API account: To collect data from Twitter, organizations need to set up a Twitter API account. This involves creating a Twitter Developer account and obtaining an API key.

Define relevant keywords: Organizations need to define relevant keywords that are related to the product, service, or topic they want to analyze. This will help to filter out irrelevant tweets.

Use a data collection tool: Organizations can use data collection tools like Twint or Tweepy to collect data from Twitter. These tools allow organizations to specify the date range, language, and number of tweets to collect.

Preprocess the data: After collecting the data, organizations need to preprocess it using techniques like tokenization, stopwords removal, and stemming to convert raw text data into a structured format that can be analyzed by a machine learning model.

1.1.4.4. Data Preprocessing

Sentiment analysis is the process of identifying and categorizing opinions expressed in text. Data preprocessing is a critical step in sentiment analysis that involves transforming raw text data into a structured format that can be analyzed by a machine learning model. In this article, we will discuss the three most common techniques for data preprocessing in sentiment analysis: tokenization, stopwords removal, and stemming.

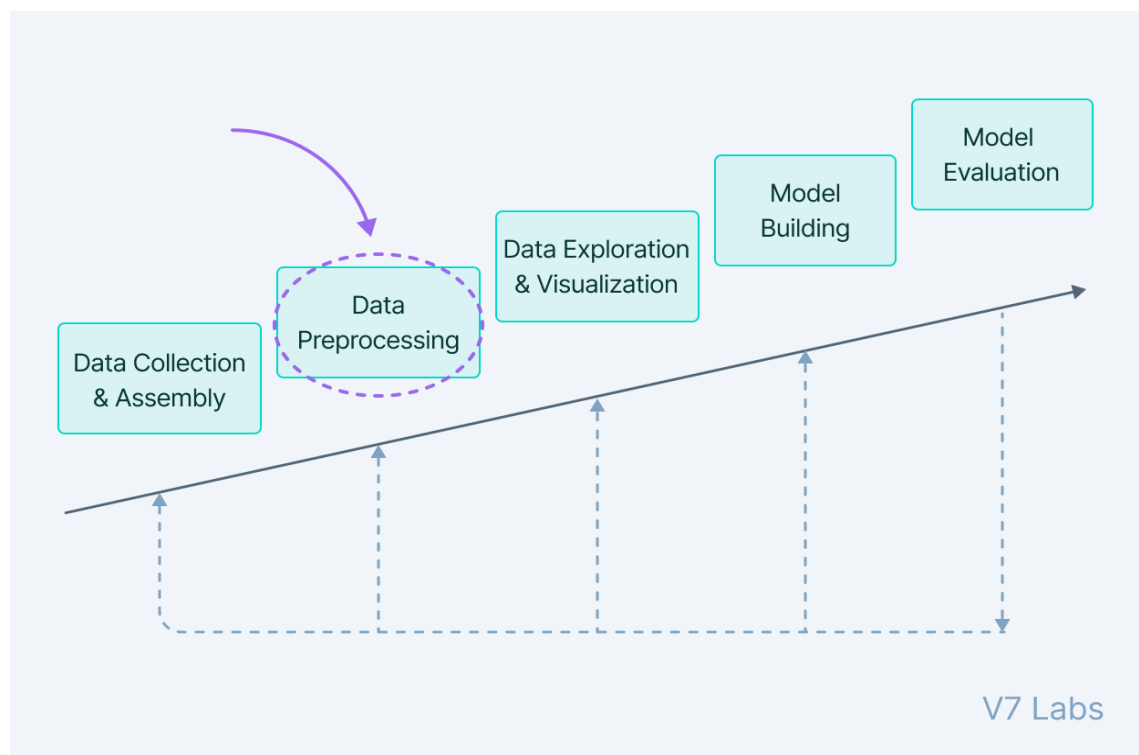


Figure I-7 Data Preprocessing[28]

I.1.4.4.1. Steps in Data Preprocessing

Now, let's discuss more in-depth four main stages of data preprocessing.

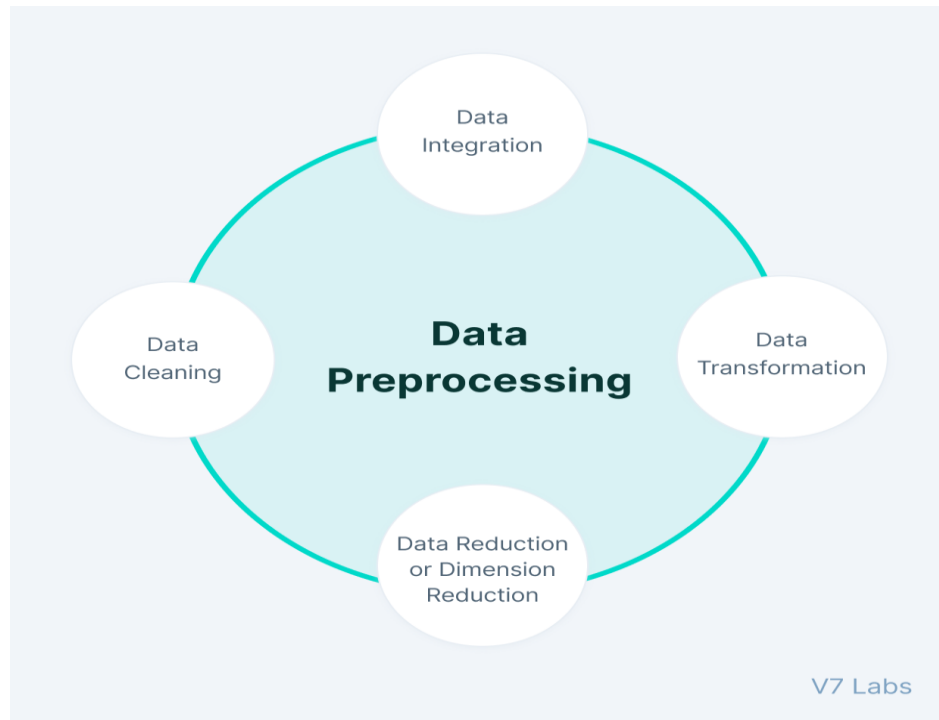


Figure I-8 : Stages of Data Preprocessing [29]

- **Data Integration**

Data integration in sentiment analysis refers to the process of combining and merging data from multiple sources to create a unified dataset. This involves dealing with challenges such as different data formats, structures, and semantics. It aims to create a comprehensive dataset that can provide a more holistic view of sentiment.

- **Data Cleaning**

Data cleaning involves preprocessing and refining the raw data to ensure its quality and remove any inconsistencies, noise, or irrelevant information. In sentiment analysis, data cleaning techniques can include removing HTML tags, punctuation, stopwords, and performing tasks like tokenization, stemming, and lemmatization.

- **Data Reduction on Dimension Reduction**

Dimension reduction techniques aim to reduce the dimensionality of the data while preserving its important features. In sentiment analysis, dimension reduction methods can help handle high-dimensional data by selecting relevant features or transforming the data into a lower-dimensional space. Common techniques include Principal Component Analysis (PCA) and t-SNE (t-Distributed Stochastic Neighbor Embedding).

- **Data Transformation**

Data transformation involves modifying the data to enhance its quality, usability, or representation. In sentiment analysis, data transformation techniques can include converting text to numerical representations using methods like word embeddings (e.g., Word2Vec, GloVe) or applying sentiment lexicons to assign sentiment scores to words or phrases.

- **Tokenization:**

Tokenization involves breaking down a sentence or document into individual words or tokens. This helps simplify the data and makes it easier to analyze. For example, the sentence "I love this movie" can be tokenized into the individual words "I", "love", "this", and "movie".

- **Stopword Removal:**

Stopwords are common words that do not carry much meaning, such as "the", "and", and "a". Removing these words can help reduce noise in the data and improve the accuracy of sentiment analysis. For example, the sentence "The movie was good, but the ending was bad" can be processed by removing the stopwords "the" and "was", resulting in "movie good ending bad".

- **Stemming:**

Stemming involves reducing words to their base or root form, such as "running" to "run". This helps normalize the data and reduce the number of unique words in the dataset. For example, the words "running", "runs", and "ran" can all be stemmed to "run".

I.2. Explaining deep learning models

I.2.1. Introduction

Deep learning models have become increasingly popular for analyzing sentiment in text data. However, one challenge of using these models is that they can be difficult to interpret. This lack of interpretability can make it challenging to understand how the model makes its predictions and identify any biases or errors in the model. To address this issue, researchers have developed various approaches to explain deep learning models for sentiment analysis.[30]

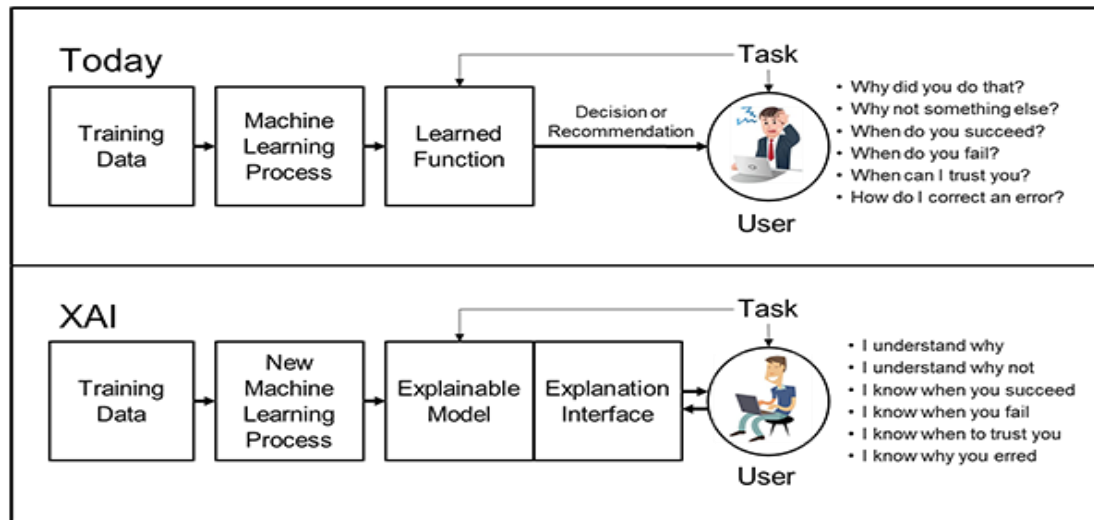


Figure I-9: The XAI Concept[30]

I.2.2. The applications of explaining deep learning models

Explaining the inner workings and decisions of deep learning models, also known as interpretability, has numerous applications across different domains. Here are some areas where explaining deep learning models is commonly used [31]–[33]:

Healthcare and Medical Diagnosis: Interpreting deep learning models helps healthcare professionals understand the reasoning behind medical diagnoses. By providing explanations for predictions made by models analyzing medical images, patient data, or genomics data, it enhances trust, assists in diagnosis, and supports treatment planning.

Autonomous Systems: Deep learning models used in autonomous systems, such as self-driving cars or drones, need to be interpretable for safety and reliability. Understanding the decisions made by these models is crucial to ensure that they operate within desired parameters and can be trusted in critical situations.

Financial Decision-Making: Interpretable deep learning models are valuable in finance and investment applications. By explaining the factors influencing predictions related to stock market trends, risk assessments, credit scoring, or fraud detection, they provide insights to financial professionals and decision-makers, aiding in making informed choices.

Customer Behavior Analysis: Deep learning models used in customer behavior analysis, such as recommender systems or churn prediction, can benefit from interpretability. By

explaining the reasons behind recommendations or churn predictions, businesses can better understand user preferences, provide personalized experiences, and optimize marketing strategies.

Natural Language Processing (NLP): Interpretability is important in NLP tasks, such as sentiment analysis, text classification, or machine translation. By explaining the basis for sentiment or classification predictions, it helps users understand how language models interpret and process textual data, leading to better insights and improved performance.

Ethical and Legal Compliance: Interpreting deep learning models is crucial in ensuring ethical use and legal compliance. It helps detect and mitigate potential biases, identify discriminatory patterns, and ensure that models adhere to legal frameworks, such as data protection and fairness regulations.

Image and Video Understanding: In computer vision tasks, such as object detection, image segmentation, or video analysis, interpretability plays a significant role. It helps understand how models interpret visual features, identify objects or regions of interest, and make predictions, enabling applications in surveillance, robotics, and medical imaging.

Research and Model Improvement: Interpretability aids researchers in understanding and improving deep learning models. By providing insights into the model's decision-making process, it helps identify model weaknesses, potential errors, or limitations, leading to advancements in architecture design, training methodologies, or data preprocessing techniques.

These are just a few examples of how explaining deep learning models can be applied across various domains. Interpretable models foster trust, enable better decision-making, and facilitate understanding of complex systems, leading to safer, fairer, and more reliable AI applications.

I.2.3. Visualize Deep Learning Models

Visualization of deep learning models is an essential tool for understanding and interpreting their internal workings. Here are some common techniques used to visualize deep learning models [34]–[36]:

Activation Visualizations: Activation visualizations help understand how information flows through the layers of a deep learning model. Techniques like heatmaps or activation maps can

highlight regions in an input image that contribute to specific activations, allowing users to visualize which parts of the image are important for a given prediction.

Filter Visualization: Filter visualization techniques aim to visualize the learned filters or feature maps of convolutional layers in a deep learning model. This provides insights into the type of features the model is detecting, such as edges, textures, or object shapes. Visualizing filters can help understand what the model has learned and how it processes visual information.

Grad-CAM(Gradient-weighted Class Activation Mapping) : is a technique that produces visual explanations for the predictions made by a deep learning model. It highlights the regions in an input image that are most influential in the model's decision-making process. Grad-CAM provides intuitive visualizations that can help interpret and explain the model's reasoning .

T-SNE Visualization(t-distributed Stochastic Neighbor Embedding) : is a dimensionality reduction technique often used to visualize high-dimensional data. It can be applied to visualize the learned representations of deep learning models. T-SNE reduces the dimensionality of the learned features and plots them in a 2D or 3D space, facilitating the understanding of cluster formations or patterns in the data.

Activation Maximization: Activation maximization techniques generate synthetic input images that maximally activate specific neurons or filters in a deep learning model. By visualizing these generated images, one can gain insights into what patterns or features the model is detecting and what it considers important for specific activations.

Network Visualization: Network visualization techniques aim to visualize the architecture and connections of deep learning models. This includes visualizing the layers, nodes, and connections between them, providing an overview of the model's structure. Network visualization helps understand the flow of information and the relationships between different parts of the model.

Model Interpretation Libraries: Several libraries and tools have been developed to aid in visualizing deep learning models. Examples include TensorBoard, which provides interactive visualizations of training and model performance, and Lucid, a library focused on visualizing and understanding neural networks.

Visualization techniques for deep learning models play a vital role in understanding their behavior, identifying potential issues, and gaining insights into the learned representations. They help researchers, practitioners, and end-users interpret and trust the models, making them more accessible and facilitating their practical applications.

I.2.4. Tools of explaining deep learning models

Interpretable Model-Agnostic Explanations (LIME)

Local Interpretable Model-Agnostic Explanations (LIME) is another powerful tool for explaining deep learning models for sentiment analysis. LIME generates locally interpretable model-agnostic explanations, meaning that it can be applied to any type of model. LIME works by generating a simpler, interpretable model for a particular prediction and then using this simpler model to identify the features of the input data that are most important for that prediction. This approach allows human experts to understand how the model is making its predictions and identify any potential biases or errors[37].

It is important to note that while these approaches provide valuable tools for explaining deep learning models for sentiment analysis, they do have some limitations. For example, saliency maps and attention weights only provide a local explanation for a single prediction, and it is necessary to aggregate these explanations across multiple examples to get a more general understanding of how the model works. Additionally, LIME explanations may not always capture the full complexity of the original model, which can limit their usefulness.

It is important to note that while these approaches provide valuable tools for explaining deep learning models for sentiment analysis, they do have some limitations. For example, saliency maps and attention weights only provide a local explanation for a single prediction, and it is necessary to aggregate these explanations across multiple examples to get a more general understanding of how the model works. Additionally, LIME explanations may not always capture the full complexity of the original model, which can limit their usefulness.

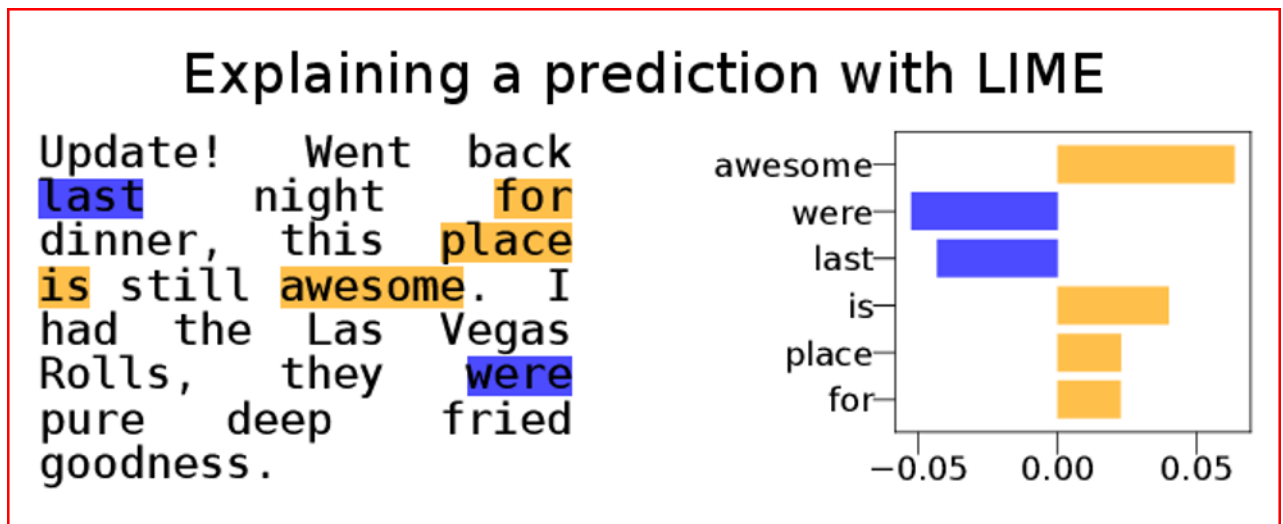


Figure I-10: Explaining the prediction of a random forest classifier on a Yelp review

It is also important to consider the ethical implications of explainability in deep learning models. Explainability could potentially reveal sensitive information about individuals or groups, leading to privacy concerns. Therefore, it is crucial to balance the benefits of transparency with the potential risks and take appropriate measures to protect privacy and confidentiality.

Explainability is also essential for ensuring regulatory compliance in some domains, such as healthcare or finance. Regulatory bodies may require models to be explainable and provide clear justifications for their predictions to ensure fair and unbiased decision-making.

1.1.1 SHAP (SHapley Additive exPlanations)

is a powerful technique that provides insights into the contributions of each feature towards a model's prediction. SHAP values assign a value to each feature, indicating its impact on the model's output. Here's an explanation of how SHAP works[38]:

Background and Shapley Values: SHAP is based on the concept of Shapley values from cooperative game theory. Shapley values quantify the marginal contribution of each player in a cooperative game. In the context of deep learning models, features act as players, and the prediction is the outcome of the game.

Individual Feature Importance: SHAP calculates the contribution of each feature for a specific instance's prediction. It considers all possible combinations of features and calculates the average marginal contribution of each feature across different permutations.

Global Feature Importance: SHAP also provides an understanding of the overall importance of features across the entire dataset. By aggregating the individual SHAP values from multiple instances, it reveals the features that have the most significant impact on the model's predictions.

Interpretation: Positive SHAP values indicate features that contribute positively to the prediction, while negative values indicate features that have a negative impact. Features with SHAP values close to zero have minimal influence.

Visualization: SHAP values can be visualized using various plots, such as summary plots or dependence plots. Summary plots show the distribution of SHAP values for each feature, providing an overview of their impact. Dependence plots visualize the relationship between a feature's value and its corresponding SHAP value, revealing how changes in a feature affect the model's output.

Model-Agnostic Approach: SHAP is model-agnostic, meaning it can be applied to any type of machine learning model, including deep learning models. It does not rely on specific model internals and can provide explanations for complex models with high-dimensional inputs.

By leveraging SHAP, researchers and practitioners can gain a better understanding of deep learning models. It helps in model validation, feature selection, and understanding the decision-making process. SHAP provides interpretable insights, enhancing trust and facilitating the deployment of deep learning models in various domains.

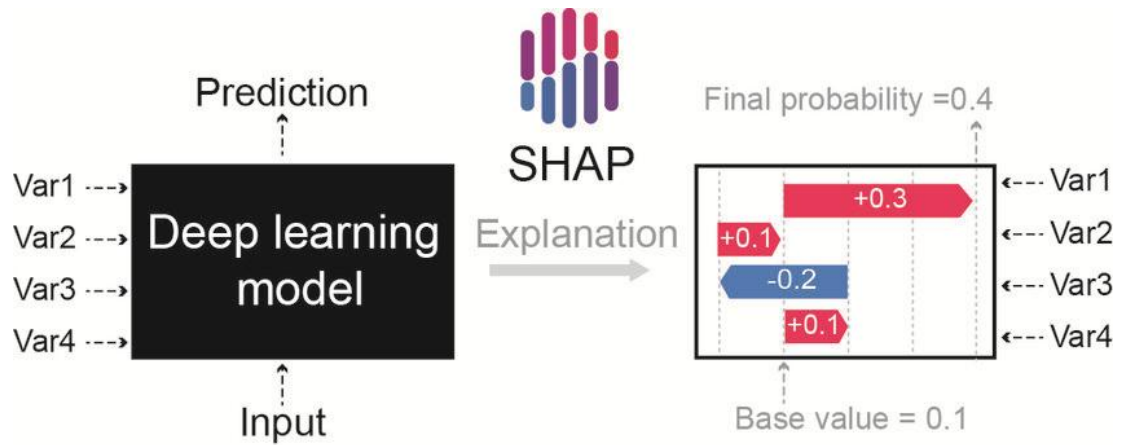


Figure I-11 Shap-explained model deep learning models[39]

Activation Maximization :

Activation maximization is used to generate input patterns that maximize the activations of specific neurons in the model. This technique can reveal what the model has learned to recognize and helps understand what features are being detected.

Layer-wise Relevance Propagation (LRP) :

LRP is a technique that propagates the relevance of the model's output back through the layers to understand which neurons or features in the input contributed most to the final prediction.

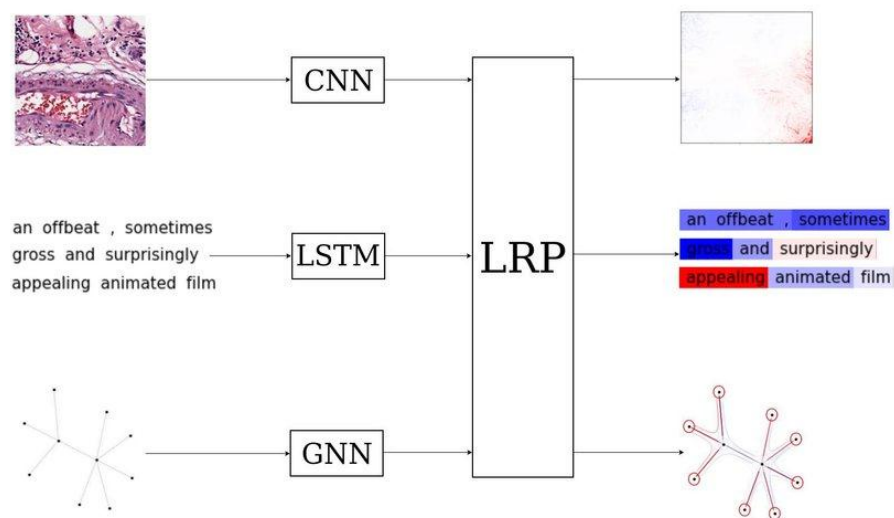


Figure I-12 Layer-wise Relevance Propagation (LRP) applied onto various neural[40]

Integrated Gradients : Integrated gradients is a method that computes the importance of each feature in the input by integrating the gradients of the model's output with respect to the input along a straight path from a reference point to the input data point.

Attention Mechanisms : Attention mechanisms in deep learning models highlight the importance of different parts of the input data during the model's decision-making process. They provide insights into the model's focus and reasoning.

Conclusion

In conclusion, sentiment analysis is a valuable technique that allows us to analyze and understand the emotions and opinions expressed in textual data. By employing various natural language processing (NLP) techniques, such as tokenization, word embedding, and machine learning algorithms, sentiment analysis enables us to classify text into positive, negative, or neutral sentiments.

Chapitre II

Analysis and conception

II. Analyse et conception

II.1. Introduction

In this chapter, we delve into a comprehensive overview of the architecture of the model proposed, the approach utilized for explaining this model, and all the techniques employed throughout the concept phase.

II.2. Definition

We are going to define some keywords in our project:

Input_Length: The maximum size of our tweet.

Trainable: Enabling the update of weights.

W_regularizer: Allows for the application of penalties on layer parameters or layer activity during optimization. These penalties are incorporated into the loss function that the network is optimizing.

Batch_Normalization: A training technique for deep neural networks that standardizes the inputs of a layer for each mini-batch. This stabilizes the learning process and reduces the number of training epochs required to train deep networks.

Epoch: The maximum number of training iterations.

Batch_Size: The number of samples propagated through the network in each training iteration.

Embedding layer: Provides a dense representation of words and their relative meanings. They can also be learned as part of fitting a neural network on textual data.

Dropout: The Dropout layer randomly sets input units to 0 with a frequency rate at each step during training, which helps prevent overfitting.

Tokenizer: A tokenizer that splits a string using a regular expression, matching either tokens or separators between tokens.

II.3. Architecture du système

Our research project revolves around sentiment analysis, aiming to enhance its performance. To achieve this, we propose the system illustrated in the following figure :

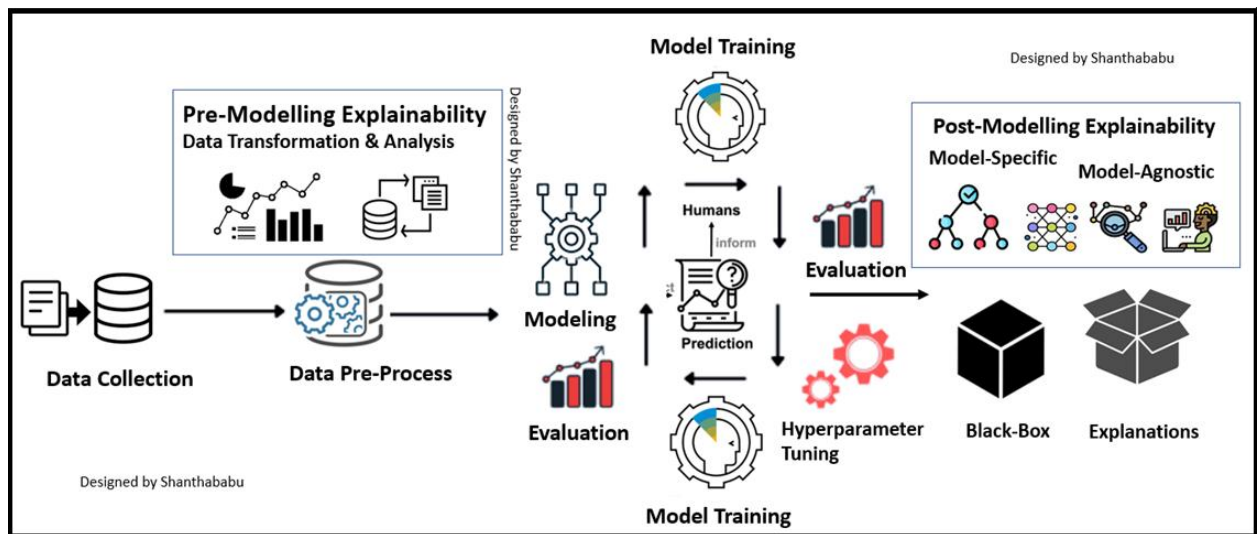


Figure II-1 Representation of the system architecture[41]

II.4. Steps of the process

In this stage, we need to clean the tweets and apply various techniques mentioned below to process the text.

- **Filtering**

The filtering step aims to remove metadata, such as URLs and symbols, from the messages.

- **Removal of stop words**

In this step, we eliminate words that do not contribute to the expressed context in the tweets. These words include determiners, conjunctions, and prepositions, which can increase the vocabulary size without providing valuable information.

- **Tokenization**

Tokenization involves segmenting the message into individual words by identifying spaces and punctuation marks. The objective is to create a bag of words representation. Special cases, such as handling apostrophes, should be taken into consideration.

- **Stemming**

Stemming is the process of reducing words to their root form. Its purpose is to group words with the same root, thereby reducing redundancy in the vocabulary caused by inflections.

- **Lemmatization**

Lemmatization is a more advanced technique than stemming. It aims to transform words to their canonical or base form, known as a lemma. Unlike stemming, lemmatization ensures that the resulting lemma is a valid word. For example, the word "étions" would be lemmatized to "être" and "cherchais" to "chercher". Lemmatization often involves referencing a dictionary to identify the correct lemma for a word.

- **Feature extraction**

Feature extraction involves converting raw data into numerical features that can be processed while retaining the original information. In our study, we perform this step using approche:

For PLMs (Pre-trained Language Models), we rely on embeddings.

II.5. Model used for classification

The Transformer is a sophisticated deep learning model architecture built for natural language processing (NLP) sequence-to-sequence applications. It features the self-attention mechanism, allowing it to successfully grasp long-range relationships in input sequences. The model is made up of an encoder and a decoder, and it learns diverse patterns using multi-head attention. The model understands the word order in sequences thanks to positional encoding. The Transformer, with its exceptional performance and adaptability, has been the foundation of several cutting-edge NLP models, changing the area of natural language processing. The Transformer is a sophisticated deep learning model architecture built for natural language processing (NLP) sequence-to-sequence applications. It features the self-attention mechanism, allowing it to successfully grasp long-range relationships in input sequences. The model is made up of an encoder and a decoder, and it learns diverse patterns using multi-head attention. The model understands the word order in sequences thanks to positional encoding. The Transformer, with its exceptional performance and adaptability, has been the foundation of several cutting-edge NLP models, changing the area of natural language processing[42].

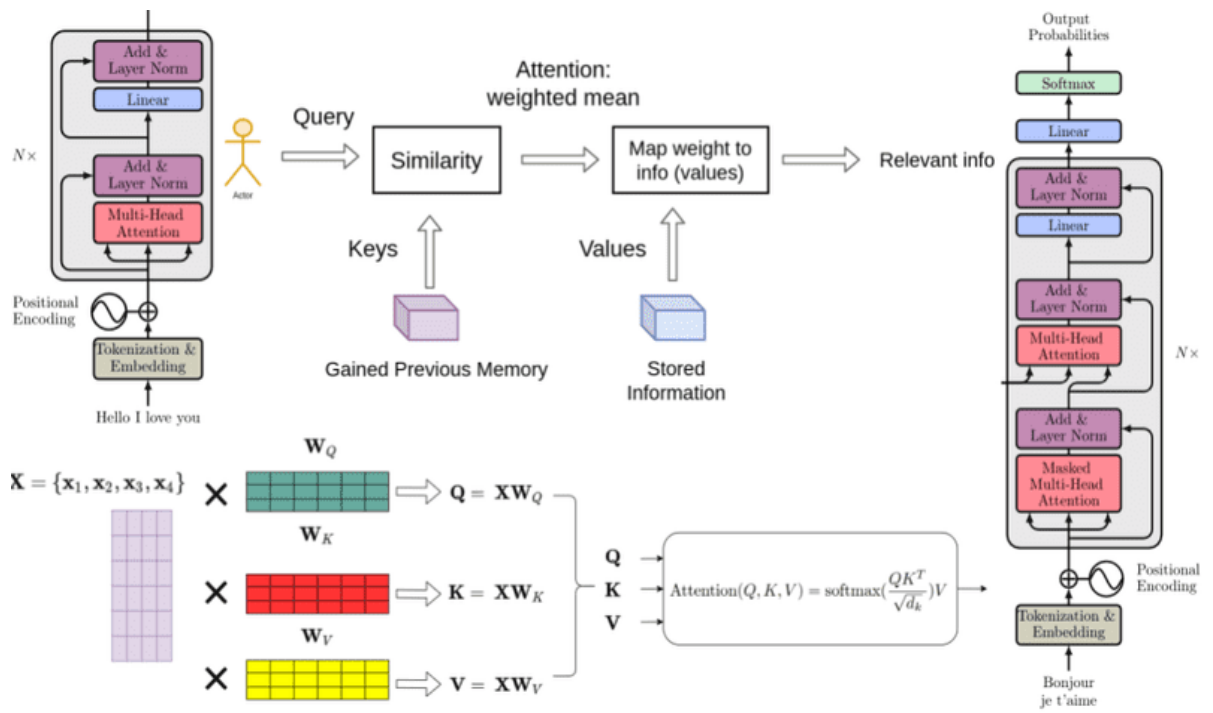


Figure II-2 How Transformers work in deep learning and NLP[43]

Pre-trained language models based on transformers have opened up a new era in the field of NLP.

After extensively studying the literature, we have been inspired to utilize and construct a model based on Pre-trained Language Models (PLMs) using the fine-tuning approach.

In our sentiment analysis classification project, we have opted as deep learning models.

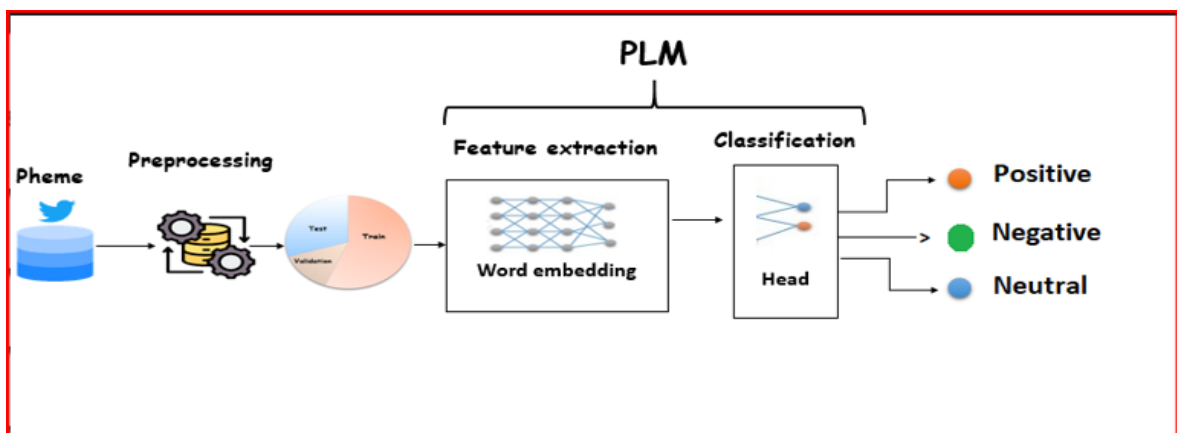


Figure II-3 Fine - tuning - based classification of a PLM[44]

II.5.1. Model01

For the first model, we have selected a fundamental classification model called **bert** with the following set of parameters :

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 80)]	0	[]
input_2 (InputLayer)	[(None, 80)]	0	[]
bert (TFBertMainLayer)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 80, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	109482240	['input_1[0][0]', 'input_2[0][0]']
dense (Dense)	(None, 32)	24608	['bert[0][1]']
batch_normalization (BatchNormalization)	(None, 32)	128	['dense[0][0]']
dense_1 (Dense)	(None, 3)	99	['batch_normalization[0][0]']
activation (Activation)	(None, 3)	0	['dense_1[0][0]']
=====			
Total params: 109,507,075			
Trainable params: 109,507,011			
Non-trainable params: 64			

Figure II-4 Configuration of model 01

II.5.2. Model02

In the second model, we applied **RoBERTa** with the following set of parameters :

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 80)]	0	[]
input_2 (InputLayer)	[(None, 80)]	0	[]
roberta (TFRobertaMainLayer)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 80, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	124645632	['input_1[0][0]', 'input_2[0][0]']
dense (Dense)	(None, 32)	24608	['roberta[0][1]']
batch_normalization (BatchNormalization)	(None, 32)	128	['dense[0][0]']
dense_1 (Dense)	(None, 3)	99	['batch_normalization[0][0]']
activation (Activation)	(None, 3)	0	['dense_1[0][0]']
=====			
Total params: 124,670,467			
Trainable params: 124,670,403			
Non-trainable params: 64			

Figure II-5 Configuration of model 02

II.6. explaining tools used

LIME(Local Interpretable Model-Agnostic Explanations)

Local Interpretable Model-Agnostic Explanations (LIME) is a framework that provides interpretability for machine learning models, regardless of their underlying algorithm. LIME achieves this by generating local explanations for individual predictions.

The key steps involved in the conception of LIME are as follows:

Perturbation: For each sampled instance, LIME creates perturbed versions of the data by randomly modifying some of its features while keeping other features fixed. The perturbations introduce local changes to the instance.

Prediction: The perturbed instances are then fed into the black-box machine learning model for prediction. The model's output for each perturbed instance is recorded.

Local Model: LIME constructs a simpler, interpretable model (e. g. , linear regression, decision tree) to approximate the behavior of the black-box model in the local neighborhood of the sampled instance. The interpretable model is trained on the perturbed instances and their corresponding predictions from the black-box model.

Feature Importance: The local model is analyzed to determine the importance of different features in making predictions. The importance of a feature is based on how much it influences the local model's output.

Explanation Generation: The feature importance values obtained from the local model are used to explain the predictions made by the black-box model. The features that have the most significant impact on the predictions are highlighted as the main contributors.

Presentation of Explanations: The explanations generated by LIME can be presented in various forms, such as textual explanations, visualizations, or feature importance rankings. These explanations provide insights into how the black-box model arrives at its predictions on a local level.

By following these steps, LIME provides model-agnostic explanations that help understand the reasoning behind individual predictions made by black-box models. It allows users to gain insights into the important features driving the model's decisions and provides transparency and trust in the model's behavior.

SHAP (SHapley Additive exPlanations):

SHAP (SHapley Additive exPlanations) is a unified framework for interpreting the predictions of machine learning models. It provides a comprehensive approach to understanding the contribution of each feature to the model's predictions. The main steps involved in the conception of SHAP are as follows:

Shapley Values: SHAP is based on the concept of Shapley values from cooperative game theory. Shapley values provide a fair way of distributing the "credit" or importance among the features in a prediction. In the context of machine learning, Shapley values assign a value to each feature that represents its contribution to the model's prediction.

Feature Importance Calculation: SHAP calculates feature importance by considering all possible combinations of features and their Shapley values. It measures the impact of including or excluding a specific feature from the model's prediction. This calculation accounts for both individual feature importance and the interactions among features.

Sample Generation: To compute the Shapley values, SHAP generates a set of samples that represent a subset of the data. These samples serve as reference points for computing feature importance. SHAP uses different sampling methods, such as Kernel SHAP or Tree SHAP, depending on the type of model being interpreted.

Explaining Predictions: Once the Shapley values are computed, SHAP provides explanations for individual predictions. It quantifies the contribution of each feature to the prediction for a specific instance. These explanations can be presented as feature importance rankings, visualizations, or textual descriptions, highlighting the relative importance of each feature in the prediction.

Global Interpretability: SHAP also allows for global interpretability by aggregating the Shapley values across all instances in the dataset. This provides insights into the overall impact of each feature on the model's predictions and helps understand the model's behavior at a higher level.

Model-Agnostic: One of the key advantages of SHAP is that it is model-agnostic, meaning it can be applied to various types of machine learning models, such as linear models, tree-based models, or deep learning models. It provides a consistent and unified approach to interpretability across different model architectures.

By following these steps, SHAP provides a powerful and flexible framework for interpreting machine learning models. It helps users understand the underlying factors influencing the model's

predictions, identify feature interactions, detect biases, and gain trust in the model's decision-making process.

Chapitre III

Implementation and discussion

III. Implementation and discussion

III.1. Introduction

The objective of the implementation phase is to first train and test our models, evaluate their accuracy, and subsequently compare them to select the most suitable model for sentiment analysis. We will provide an explanation of the models and compare their respective methods of explanation. Furthermore, we will specify the tools, programming languages, and techniques employed in this section.

III.2. The tools and libraries used

III.2.1. Langage

We have selected Python as the programming language due to its stability, flexibility, and robustness. Python offers a comprehensive standard library and supports multiple programming paradigms, including structured, object-oriented, and functional programming. It is widely used in the field of machine learning due to its flexibility in modeling and its rich set of existing functions for machine learning tasks.

III.2.2. Tools

Google Colaboratory is a cloud service provided by Google that is based on Jupyter Notebook. It is specifically designed for machine learning training and research, eliminating the need for hardware constraints. With Colab, notebooks can run code on Google's cloud servers, allowing users to leverage the power of Google's hardware, such as GPUs and TPUs, regardless of their own machine's capabilities. In our project, we utilize the Colab Pro version for our work.

III.2.3. Libraries

- **Transformers**

Transformers is a natural language processing library developed by Hugging Face with the support of a vibrant community of over 400 external contributors. It offers APIs for effortless downloading and fine-tuning of cutting-edge pretrained models.

- **Datasets**

Datasets is a library that facilitates easy access and sharing of datasets and evaluation metrics for natural language processing. It enables us to effortlessly load publicly available datasets provided on the Hugging Face Datasets Hub. Additionally, it allows efficient handling and preparation of our own datasets for inspection, evaluation, and model training.

- **Scikit-learn**

Scikit-learn is a valuable and robust library for machine learning. It provides a wide array of effective tools for various machine learning tasks, including classification, regression, clustering, and dimensionality reduction. It implements all the necessary classifiers needed for our study.

- **Matplotlib**

Matplotlib is a comprehensive library in Python for creating static, animated, and interactive visualizations. It can be combined with other Python libraries for scientific computing, such as NumPy and SciPy.

- **Seaborn**

Seaborn is a Python library for data visualization that is built on top of Matplotlib and integrates well with Pandas data structures. It offers a high-level interface for creating visually appealing and informative statistical graphics. Seaborn is equally powerful as Matplotlib but provides added simplicity and unique functionality.

- **Pandas**

Pandas is an open-source library written in Python specifically designed for data manipulation and analysis. It offers data structures and operations for handling numerical arrays and time series data, making it a powerful tool for data exploration and manipulation.

Configuration Used in implementation

The hardware configuration used in our implementation is :

- ❖ Un PC portable Dell Pentium CPU 2.50 GHZ
- ❖ RAM de taille 4 GO
- ❖ Disque dur de taille 500 GO

❖ Système d'exploitation Ubuntu Linux

III.3. Datasat

We have used the entire dataset of tweets labeled by the SemVal2017. It is represented in a .txt file containing 62K tweets . This dataset consists of three classes: positive, negative, and neutre. Each entry in our dataset is structured as follows:

Sure, here's the reformulated version :

ID : An identifier for the tweet.

Tweet : The text posted by the user in the tweet.

Class : The label assigned to the tweet, which can be "positive", "negative", or "neutre".

We have divided our dataset into three parts: train, validation, and test.

Certainly! Here's the reformulated version:

a) Training dataset:

This type of data is used to train a deep learning algorithm. The data scientist provides input data to the algorithm along with the corresponding expected output. The model iteratively processes the data to understand its patterns and behavior, and adjusts its parameters to optimize its performance for the intended task.

b) Validation dataset:

During the training process, the validation data is used to assess the performance of the model on unseen data. It serves as a checkpoint to evaluate how well the model generalizes to new data and helps in tuning hyperparameters. Data scientists can analyze the model's predictions on the validation dataset to make improvements and prevent overfitting.

c) Test dataset:

Once the model is trained and fine-tuned using the training and validation datasets, it is evaluated on the test dataset. This dataset consists of unseen data that the model has not been

exposed to during training. The test dataset provides an unbiased assessment of the model's performance, verifying its ability to make accurate predictions in real-world scenarios. It helps validate the generalization capabilities of the model before deploying it for practical applications.

The data distribution is illustrated in Table I-1 after preprocessing. In this step, we rename the classes "positive," "negative," and "neutre" to 0, 1, and 2, respectively.

Dataset	0 :neutral	1 :positive	2 :negative	Total
Tweet	28461	22174	11781	62416

Tableau III-1Dataset description SemVal2017

III.4. Results :

III.4.1.Evaluation metrics

- **Accuracy:** It is a metric that evaluates the overall performance of a model across all classes, assuming equal importance for each class. It is calculated as the ratio of the number of correct predictions to the total number of predictions.
- **Recall:** Also known as sensitivity or true positive rate, it measures the proportion of positive samples correctly identified by the model out of the total number of actual positive samples. Recall indicates the model's ability to detect positive samples, and a higher recall value suggests a higher number of correctly identified positive samples.
- **F1-score:** The F1-score combines precision and recall into a single metric, specifically for a particular positive class. It is the harmonic mean of precision and recall, providing a balanced evaluation of the model's performance. The F1-score ranges from 0 to 1, with a higher score indicating better performance, where 1 is the best and 0 is the worst possible value.

III.4.2. Implementation

Model 01 :

For our first model, we used a transformer model: **BertModel**

Code source :

```
# This is the best model
def Offense_classifier(input_shape):

    model = TFBertModel.from_pretrained('bert-base-uncased')
    layer = model.layers[0]

    # Define sentence_indices as the input of the graph, it should be of shape input_shape and dtype 'int32' (as it contains indices).
    inputs = keras.Input(shape=input_shape, dtype='int32')
    input_masks = keras.Input(shape=input_shape, dtype='int32')

    embeddings = layer([inputs, input_masks])[1]

    X = keras.layers.Dense(32,activation='elu',kernel_regularizer=keras.regularizers.l2(0.0001))(embeddings)

    X = keras.layers.BatchNormalization(momentum=0.99, epsilon=0.001, center=True, scale=True)(X)

    X = keras.layers.Dense(3,activation='tanh',kernel_regularizer=keras.regularizers.l2(0.0001))(X)

    # Add a sigmoid activation
    X = keras.layers.Activation('softmax')(X)

    # Create Model instance which converts sentence_indices into X.
    model = keras.Model(inputs=[inputs,input_masks], outputs=[X])

    return model
```

Results :

	precision	recall	f1-score	support
0	0.758	0.525	0.620	5937
1	0.607	0.760	0.675	2375
2	0.640	0.838	0.726	3972
accuracy			0.672	12284
macro avg	0.668	0.708	0.674	12284

Tableau III-2 Results of model BERT

Note:

In this table, we notice that the training accuracy is 67%, with a precision of 75% for class 0, based on a support of 5937 texts and 60 % for class 1 of 2375 texts and 64 % for class 2 of 3972 .

Model 02 :

For our second model, we used a translation model : **RoBERTa**

Code source :

```
from transformers import TFRobertaModel

def Offense_classifier(input_shape):

    model = TFRobertaModel.from_pretrained('roberta-base')
    layer = model.layers[0]

    # Define sentence_indices as the input of the graph, it should be of shape input_shape and dtype 'int32' (as it contains indices).
    inputs = keras.Input(shape=input_shape, dtype='int32')
    input_masks = keras.Input(shape=input_shape, dtype='int32')

    embeddings = layer([inputs, input_masks])[1]

    X = keras.layers.Dense(32, activation='elu', kernel_regularizer=keras.regularizers.l2(0.0001))(embeddings)

    X = keras.layers.BatchNormalization(momentum=0.99, epsilon=0.001, center=True, scale=True)(X)

    X = keras.layers.Dense(3, activation='tanh', kernel_regularizer=keras.regularizers.l2(0.0001))(X)

    # Add a sigmoid activation
    X = keras.layers.Activation('softmax')(X)

    # Create Model instance which converts sentence_indices into X.
    model = keras.Model(inputs=[inputs, input_masks], outputs=[X])

    return model
```

Results :

	precision	recall	f1-score	support
0	0.779	0.523	0.626	5937
1	0.584	0.796	0.673	2375
2	0.656	0.835	0.735	3972
accuracy			0.677	12284
macro avg	0.673	0.718	0.678	12284
weighted avg	0.701	0.677	0.670	12284

Tableau III-3 Results of model RoBERTa**Note:**

In this table, we notice that the training accuracy is 70%, with a precision of 77% for class 0, based on a support of 5937 texts and 58 % for class 1 of 2375 texts and 65 % for class 2 of 3972 .

III.4.3. Discussions

Model 01 :

Epoch=4

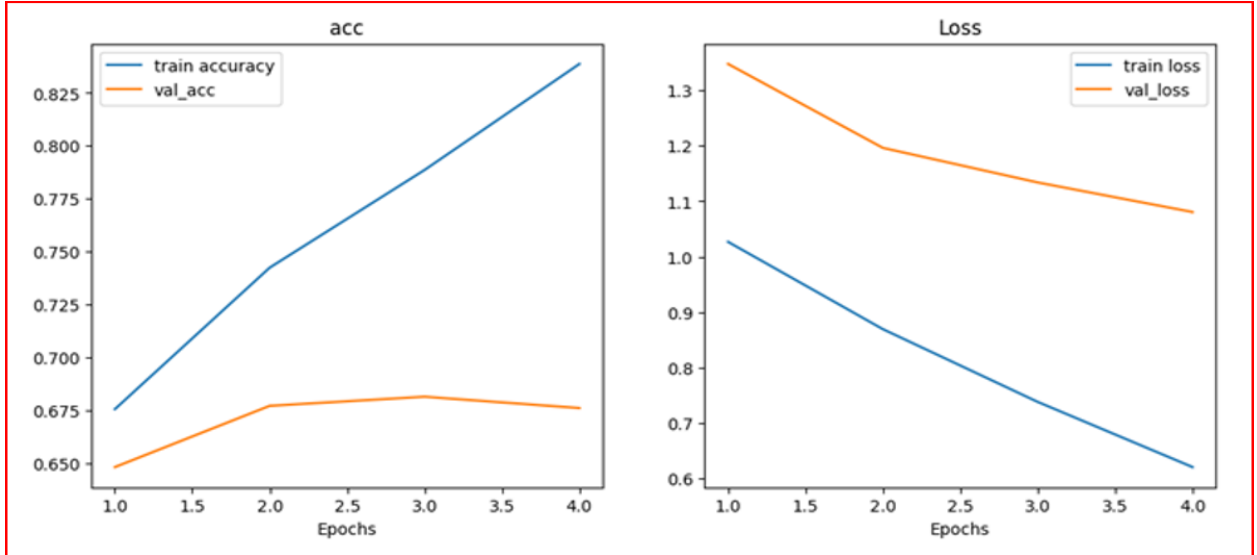


Figure III-1 Accuracy and Error for the Bert epoch=4

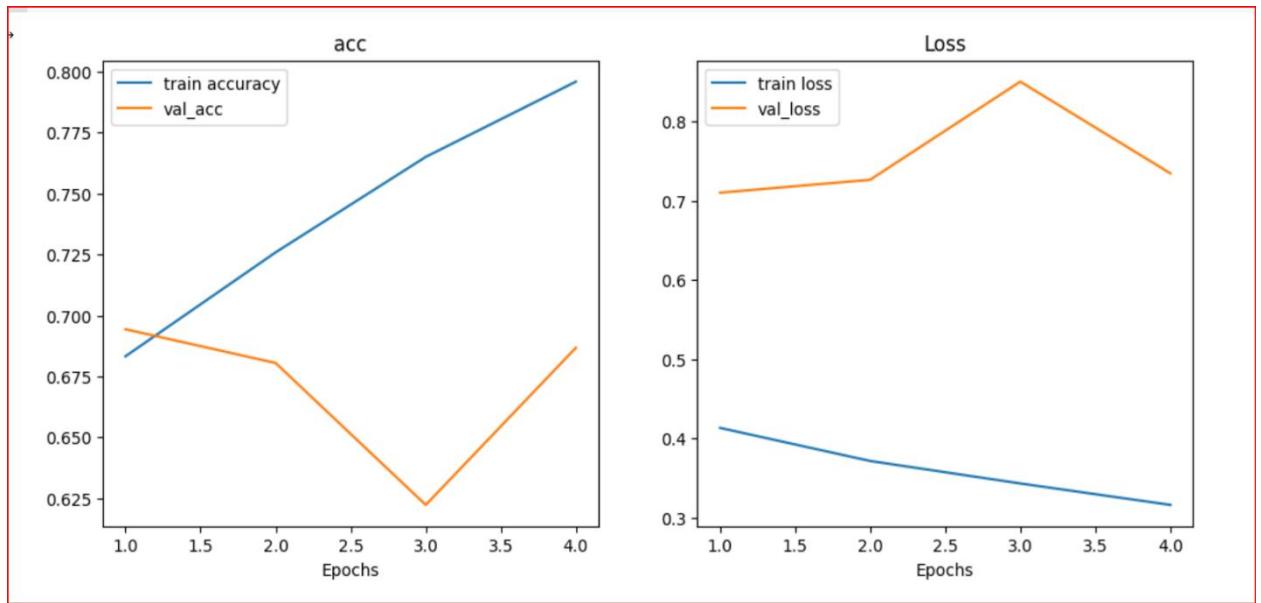
Notice:

- "Train accuracy" increases steadily up to epoch 4, indicating that the model's accuracy on the training data improves as it goes through more epochs.
- "Val_acc" (Validation accuracy) increases up to epoch 2, which means that the model's accuracy on the validation data improves up to this point. However, from epoch 2 to epoch 4, it shows stability, meaning there is no significant improvement or decline in accuracy on the validation set during these epochs.
- In "Figure 2, " the "Train loss" decreases steadily up to epoch 4, indicating that the model's loss on the training data decreases as it learns from more data during these epochs.
- The "Val_loss" (Validation loss) follows a similar trend as the "Train loss" until epoch 4, where it reaches its minimum value. This suggests that the model achieves its lowest loss on the validation data at epoch 4.

The observations indicate the following:

- The model shows steady improvement in training accuracy as it goes through more epochs, which suggests that it is effectively learning from the training data.
- The validation accuracy increases up to epoch 2, indicating that the model performs well on the validation set during the initial training phase. However, from epoch 2 to epoch 4, the validation accuracy stabilizes, implying that the model may have reached its optimal performance on the validation data.
- The training loss steadily decreases up to epoch 4, demonstrating that the model is effectively minimizing its training loss and fitting the training data better with each epoch.
- The validation loss follows a similar trend as the training loss until epoch 4, where it reaches its minimum value. This suggests that the model achieves its best performance on the validation data at epoch 4.

In conclusion, the model's training accuracy, as well as its ability to fit the training data, improves consistently with more epochs. However, the validation accuracy and loss indicate that the model's performance plateaus after epoch 2 and remains stable from epoch 2 to epoch 4. This implies that further training beyond epoch 4 may not result in significant improvements and could potentially lead to overfitting. Epoch 4 seems to be a critical point where the model achieves a good balance between generalization and fitting the data, making it a suitable choice for deployment.

Model 02 :**Figure III-2 Accuracy and Error for the RoBERTa epoch=4****Notice :**

- "Train accuracy" steadily increases with each epoch, indicating that the model's accuracy on the training data improves consistently as it learns from more data.
- "Val_acc" (Validation accuracy), on the other hand, initially decreases until epoch 2. This suggests that the model's performance on the validation data worsens during the first two epochs.
- The contradiction occurs at epoch 3, where the "Val_acc" reaches its lowest point. This means that the validation accuracy decreases to its minimum value at epoch 3, creating a sharp contradiction with the increasing trend of the training accuracy.
- After epoch 3, there is a significant and rapid increase in "Val_acc" from epoch 3 to epoch 4. This indicates that the model's performance on the validation data starts to improve rapidly after the lowest point observed at epoch 3.
- Concerning the second figure, "train loss" decreases continuously up to epoch 4, indicating that the model's loss on the training data decreases as it learns from more epochs.

- The "val_loss" (Validation loss) initially increases until epoch 3, reaching its peak at that point. This shows that the model's performance on the validation data worsens until epoch 3.
- After epoch 3, the "val_loss" starts to decrease from epoch 3 to epoch 4, suggesting that the model's performance on the validation data improves during this period.

The observations suggest the following conclusions:

- The model experiences some initial overfitting on the validation data, as indicated by the decrease in "Val_acc" and the increase in "val_loss" up to epoch 3. This means that the model's performance on unseen validation data worsens during the first few epochs.
- After epoch 3, the model starts to generalize better, leading to an increase in "Val_acc" and a decrease in "val_loss." This indicates that the model's performance on the validation data improves after epoch 3.
- Epoch 4 appears to be a critical point where the model achieves a better balance between underfitting and overfitting. At this point, both "Val_acc" and "val_loss" reach their best values. This suggests that further training beyond epoch 4 may not significantly improve the model's performance on the validation data.
- The model's performance on the training data, as indicated by "Train accuracy" and "Train loss," steadily improves with each epoch. This suggests that the model is learning and generalizing well on the training data.
- To optimize the model's performance and avoid overfitting, it may be beneficial to consider early stopping at epoch 4 or use other regularization techniques.

In summary, the observations reveal that the model's performance on the validation data improves significantly after the initial overfitting phase. Epoch 4 seems to be a turning point where the model achieves a good balance between underfitting and overfitting, leading to improved performance on unseen data. Further analysis and fine-tuning may be necessary to optimize the model's performance and achieve even better results on unseen data.

Comparison between two models

Both BERT and RoBERTa models show steady improvement in training accuracy and effective learning from the training data.

Both models initially experience some overfitting on the validation data, as indicated by a decrease in validation accuracy and an increase in validation loss during the first few epochs.

However, the BERT model's validation accuracy stabilizes after epoch 2, whereas the RoBERTa model's validation accuracy continues to improve significantly after epoch 2.

The RoBERTa model seems to perform better than the BERT model in terms of generalization, as it achieves higher validation accuracy and lower validation loss on unseen data.

Overall, the comparison suggests that the RoBERTa model outperforms the BERT model in terms of generalization and achieving better performance on unseen data. It shows more consistent improvement in validation accuracy and lower validation loss compared to the BERT model. However, further analysis and experimentation would be required to make more definitive conclusions about the superiority of one model over the other in specific tasks or domains.

Apply LIME in our Model

We will review some estimations of our model and attempt to use Lime :

twitter-2017test.txt	
1 801989080477154944	neutral #ArianaGrande Ari By Ariana Grande 80% Full https://t.co/ylhCMETHHw #Singer #Actress https://t.co/lTrb1JQiEA
2 801989272341453952	positive Ariana Grande KIIS FM Yours Truly CD listening party in Burbank https://t.co/ClQIcx8Z6V #ArianaGrande
3 801990978424962944	positive Ariana Grande White House Easter Egg Roll in Washington https://t.co/jdjl9swWM8 #ArianaGrande
4 801996232553963008	positive #CD #Musics Ariana Grande Sweet Like Candy 3.4 oz 100 ML Sealed In Box 100% Authentic New https://t.co/ofmp0bc
5 801998343442407040	neutral SIDE TO SIDE @arianagrande #sidetoside #arianagrande #musically #comunidadgay #lgbt #LOTB... https://t.co/
6 802001659970744064	positive Hairspray Live! Previews at the Macy's Thanksgiving Day Parade! https://t.co/GaFTqIno1L #arianagrande #televi
7 802003380973568000	positive #LindsayLohan Is 'Feeling Thankful' After Blasting #ArianaGrande For Wearing 'Too Much... https://t.co/Acf8ogv5
8 802014830467174016	neutral I hate her but... I love her songs Dammit... #ArianaGrande
9 802020578609623040	neutral Ariana Grande [Right There ft. Big Sean] #アリアナ #arianagrande https://t.co/4rZOKYDnUt
10 802021059490934016	positive which one would you prefer to listen to for a whole day? 🎧 ; i could never choose #arianagrande #IntoYou #
11 802021122384523008	neutral Booty Baby Ari #ArianaGrande #PrincessAri #bootybaby #DangerousWomanTour #DangerousWoman... https://t.co/YBspYl06
12 802022296818880000	neutral #LindsayLohan backs out of a #Kettering holiday appearance, just after throwing shade at #ArianaGrande. #LiLoht
13 802022559520673024	positive My idols are #Littlemix #justinbieber #arianagrande
14 802022965743415040	neutral Ariana Grande - The Sims 3 - SimsDomination #ArianaGrande https://t.co/l2K2ihUpCg https://t.co/W7sIK84nOT
15 802024079851069056	neutral #Music #ArianaGrande-THE REMIX-JAPAN ONLY #CD E78 https://t.co/z0xb3hwBpg #Bestseller #Hit https://t.co/ibZSc6
16 802024085777629056	positive #Beauty #ArianaGrande-CHRISTMAS & CHILL-JAPAN ONLY #CD BONUS TRACK C94 https://t.co/pDp9gEqAW6 #Deals #TopSe
17 802024981638958976	neutral [Popular Song] Ahh, I said I'm putting down my story in a popular song #Taylor #ArianaGrande #Justin #music
18 802026442687217024	neutral Ariana Grande Private Event for Coach in Japan, August 2015 https://t.co/5DCmT5HkrF #ArianaGrande
19 802028835013206016	positive #Beauty #ArianaGrande-THE REMIX-JAPAN ONLY #CD E78 https://t.co/ga2pApWgaK #Deals #TopSeller https://t.co/4ag
20 802028852180462976	positive 4. One last time - #ArianaGrande https://t.co/0o4T0q3GsY
21 802030814246489984	positive More #newarrivals #pentatonix #christmasalbum #arianagrande #frankzappa #leonardcohen with Hallelujah #bobby
22 802032149885026048	positive so much love for this woman, ughh 🥰🥰 #arianagrande https://t.co/3ft3LuLsQt
23 802032523152718976	neutral Ariana lips 🍷 #mac #ariana #arianagrande https://t.co/T4aK8LagQX
24 802041485948555008	neutral Ariana Grande - 2015 NYC Pride Dance On The Pier https://t.co/9xelKJuPOe #ArianaGrande
25 8020428420969902016	positive thanks God it's Friday. #Thanksgiving #arianagrande #arianator https://t.co/xx4LfmlBoz
26 802045545992723968	neutral #MUSIC #ArianaGrande@karinrino PM16:00Ariana Grande - Baby I https://t.co/VY7fmp6RbY
27 802045726318587008	positive New on @Twitter . Big fan of @NICKIMINAJ and @ArianaGrande #ArianaGrande #NickiMinaj #Barbies #Barbz #Arianat
28 802198774114385024	negative Soros brainwashes & enslaves U #blacklivesmatter protesting fools. So naive. U want socialism? Study Venezu
29 802198896474803968	neutral Obama's latest #BlackLivesMatter recruiting videowelcome to the Modesto mall... https://t.co/75W39COWAF #MAGA

Figure III-3 Part of the dataset testing

Code source :

```
import lime
from lime import lime_tabular
import lime
from lime import lime_text

# Load the model weights
# model.load_weights("/content/drive/MyDrive/master/twitter_sentiment_analysis/results/neutro-b
```

Convert test sentences into a list

```
[ ] X_val = list(X_val)
    print(X_val[0:10])
```

```
[' ariana grande  ari by ariana grande  full  singer  actress ', 'ariana grande  kiis fm your
```

Create a Lime explain

```
[ ] explainer = lime_text.LimeTextExplainer(class_names = ["0", "1", "2"]
    )
```

**Determine the model and required parameters for the Lime model **

```
def predict_proba(texts):
    encoded_texts = tokenizer(texts, max_length=80, truncation=True, padding="max_length", return_tensors='tf')
    predictions = model.predict([encoded_texts["input_ids"], encoded_texts["attention_mask"]])
    return predictions

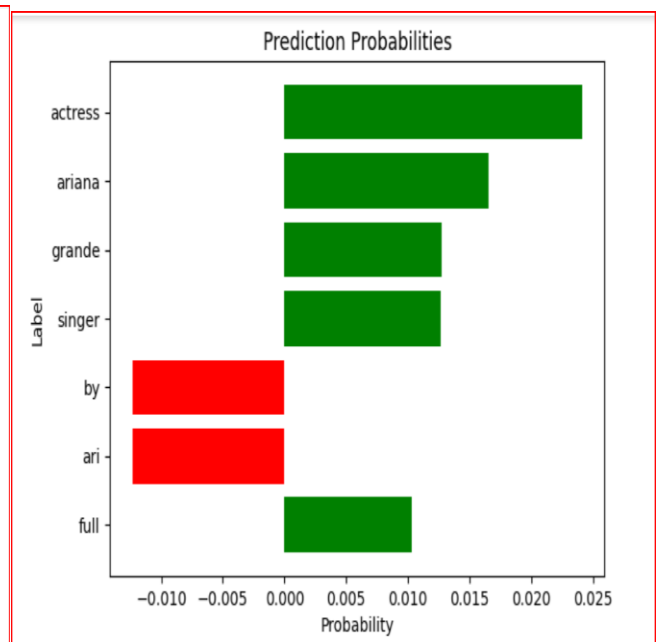
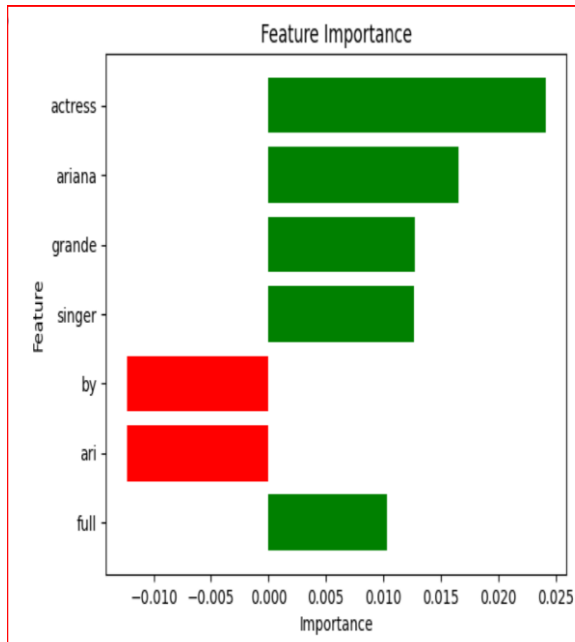
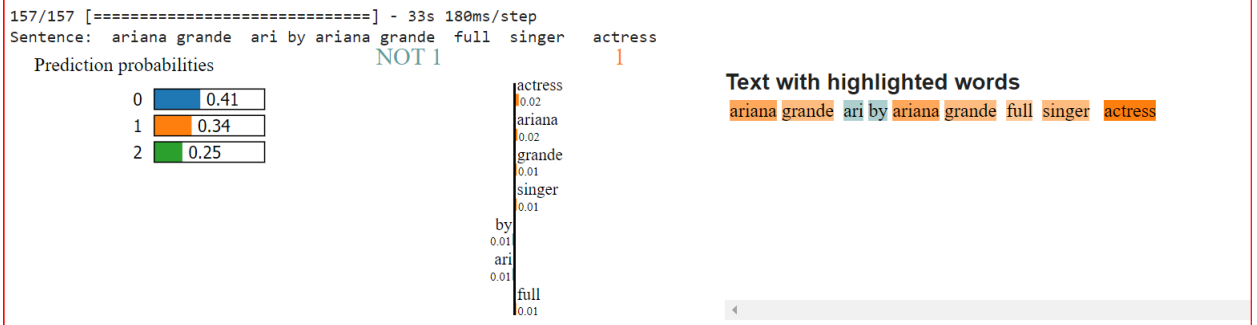
def predict_fn(texts):
    probabilities = predict_proba(texts)
    return probabilities
```

Execution result :

Model 01 : Bert

Tweet 01 : « **neutral** » #ArianaGrande Ari By Ariana Grande 80% Full

<https://t.co/ylhCMETHHW> #Singer #Actress



Notice:

After analyzing the sentence "ariana grande ari by ariana grande full singer actress" using the BERT model and interpreting it with LIME, the results of the analysis are as follows:

The original model's prediction for the sentence is:

0.41 probability for class 0 (neutral)

0.34 probability for class 1 (positive)

0.25 probability for class 2 (negative)

As for the LIME model's prediction (local explanation), it highlights the key words that influenced the sentence's classification. These words are:

"actress" with a weight of 0.02 ; "ariana" with a weight of 0.02

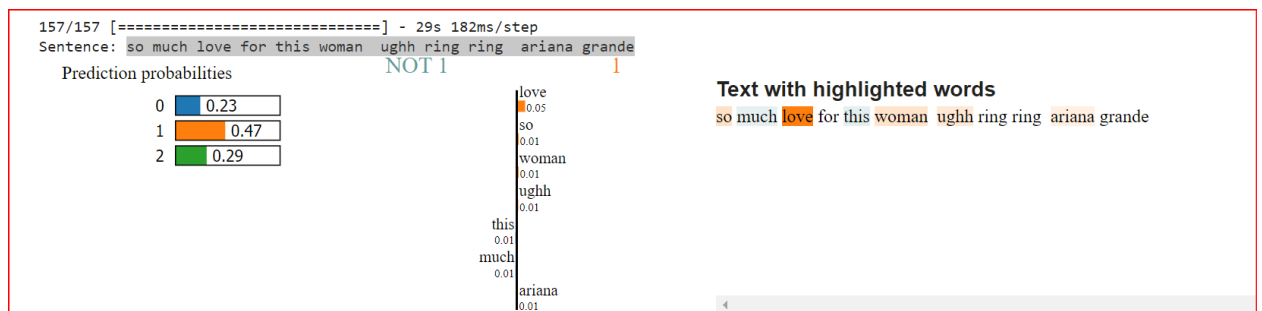
"grande" with a weight of 0.01 ; "singer" with a weight of 0.01

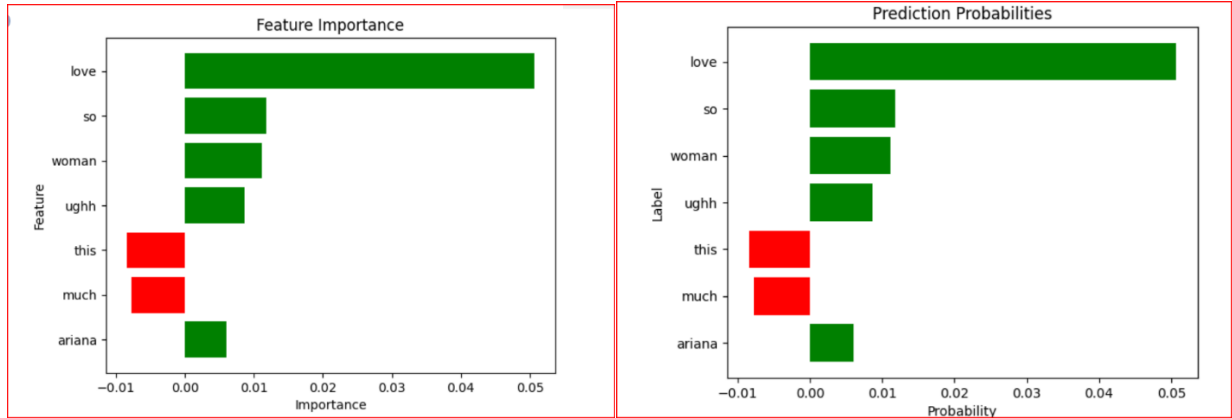
"by" with a weight of 0.01 ; "ari" with a weight of 0.01

"full" with a weight of 0.01

These words indicate the important elements that contributed to the final classification of the sentence by the LIME model. It can be inferred that the presence of words like "actress," "ariana," and "grande" enhances the probability of class 0 (neutral), while the presence of words like "singer" and "by" enhances the probability of class 1 (positive). Additionally, the word "full" has a slight and unspecified influence on the classification.

Tweet22: « **positive** » so much love for this woman, ughh ☐☐ #arianagrande



**Notice:**

After analyzing the sentence "so much love for this woman ughh ring ring ariana grande" using the BERT model and interpreting it with LIME, the results of the analysis are as follows:

The original model's prediction for the sentence is:

0.23 probability for class 0 (neutral)

0.47 probability for class 1 (positive)

0.29 probability for class 2 (negative)

As for the LIME model's prediction (local explanation), it highlights the key words that influenced the sentence's classification. These words are:

"love" with a weight of 0.05 ; "so" with a weight of 0.01

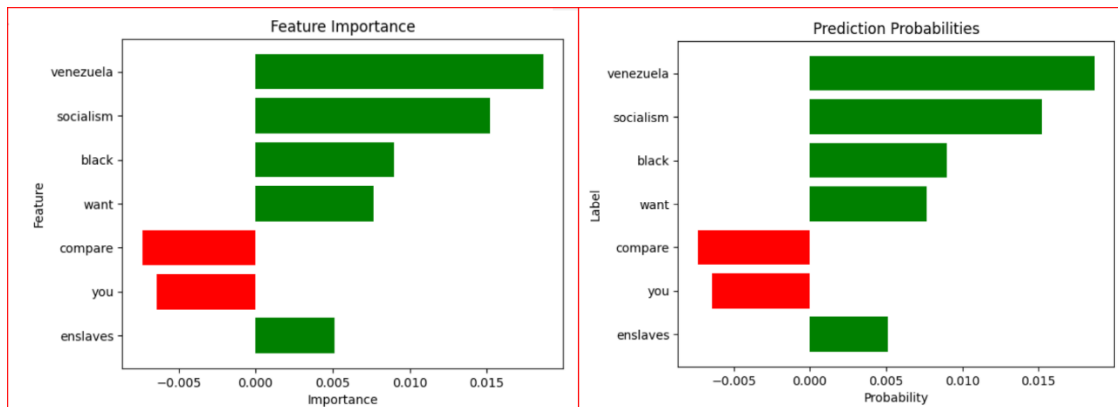
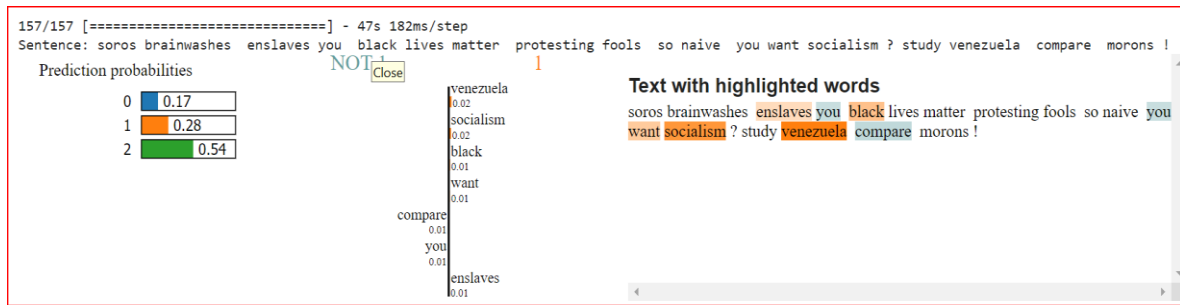
"woman" with a weight of 0.01 ; "ughh" with a weight of 0.01

"this" with a weight of 0.01 ; "much" with a weight of 0.01

"ariana" with a weight of 0.01

These words indicate the important elements that contributed to the final classification of the sentence by the LIME model. It can be inferred that the presence of the word "love" strongly enhances the probability of class 1 (positive). The other words have a lesser and unspecified influence on the classification.

Tweet 28 : : « **negative** » Soros brainwashes & enslaves U #blacklivesmatter protesting fools. So naive. U want socialism? Study Venezuela & compare. Morons!



Notice:

After analyzing the sentence "soros brainwashes enslaves you black lives matter protesting fools so naive you want socialism? study venezuela compare morons!" using the BERT model and interpreting it with LIME, the results of the analysis are as follows:

The original model's prediction for the sentence is:

0.17 probability for class 0 (neutral)

0.28 probability for class 1 (positive)

0.54 probability for class 2 (negative)

As for the LIME model's prediction (local explanation), it highlights the key words that influenced the sentence's classification. These words are:

"socialism" with a weight of 0.02 ; "venezuela" with a weight of 0.02

"black" with a weight of 0.01 ; "want" with a weight of 0.01

"compare" with a weight of 0.01 ; "you" with a weight of 0.01

"enslaves" with a weight of 0.01

These words indicate the important elements that contributed to the final classification of the sentence by the LIME model. It can be observed that words related to negative associations like "socialism," "venezuela," and "enslaves" have higher weights, which contribute to the higher probability of class 2 (negative). On the other hand, words like "want" and "compare" have a lesser and unspecified influence on the classification.

Resultas :

After analyzing the sentences using the BERT model and interpreting them with LIME, the following conclusions can be drawn:

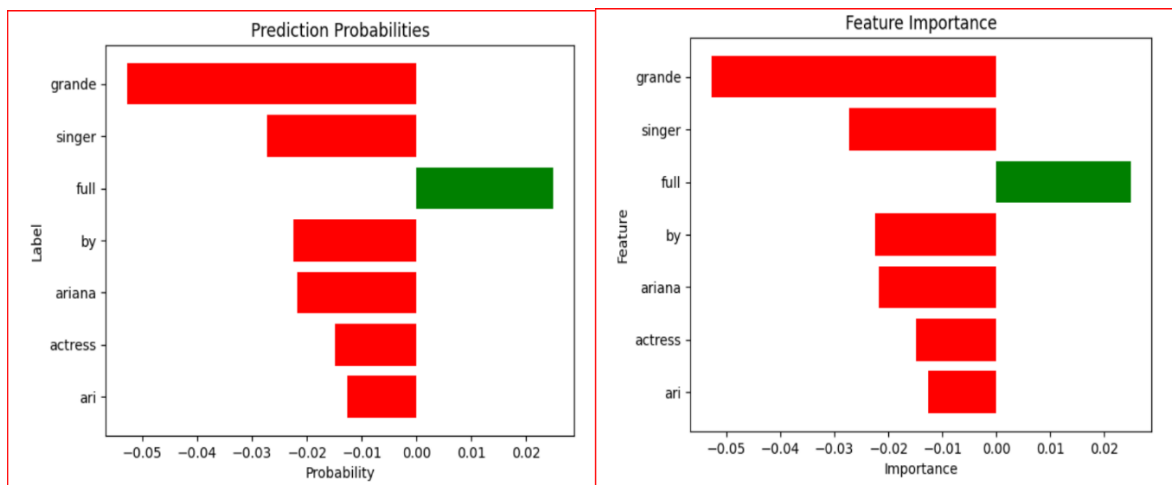
- In the sentence "ariana grande ari by ariana grande full singer actress," the LIME model highlights the key words that influenced the classification. The presence of words like "actress," "ariana," and "grande" enhances the probability of class 0 (neutral), while the presence of words like "singer" and "by" enhances the probability of class 1 (positive). The word "full" has a slight and unspecified influence on the classification. Overall, the LIME model provides insights into the important elements contributing to the classification of the sentence.
- In the sentence "so much love for this woman uhhh ring ring ariana grande," the LIME model identifies the word "love" as a key influencer, strongly enhancing the probability of class 1 (positive). The other words, such as "so," "woman," "ughh," and "ariana," have lesser and unspecified influences on the classification. The LIME model helps understand the factors contributing to the sentiment expressed in the sentence.
- In the sentence "soros brainwashes enslaves you black lives matter protesting fools so naive you want socialism? study venezuela compare morons!" the LIME model highlights words related to negative associations, such as "socialism," "venezuela," and "enslaves," with higher weights. These words contribute to the higher probability of class 2 (negative) in the original model's prediction. Words like "want" and "compare" have lesser and unspecified influences on the classification. The LIME model provides insights into the important elements that contribute to the sentiment and topics expressed in the sentence.

Overall, the LIME interpretation helps identify the key words that influenced the classification and provides insights into the factors contributing to the sentiment expressed in the sentences. It helps understand the specific words and their weights in influencing the predictions of the original model.

Model 02 :RoBERTa

Tweet 01: « **neutral** » #ArianaGrande Ari By Ariana Grande 80% Full

<https://t.co/ylhCMETHHW> #Singer #Actress



After analyzing the sentence "ariana grande ari by ariana grande full singer actress" using the ROBERTA model and interpreting it with LIME, the results of the analysis are as follows:

The original model's prediction for the sentence is:

0.78 probability for class 0 (neutral) ;

0.11 probability for class 1 (positive) ;

0.11 probability for class 2 (negative)

As for the LIME model's prediction , it highlights the key words that influenced the sentence's classification. These words are:

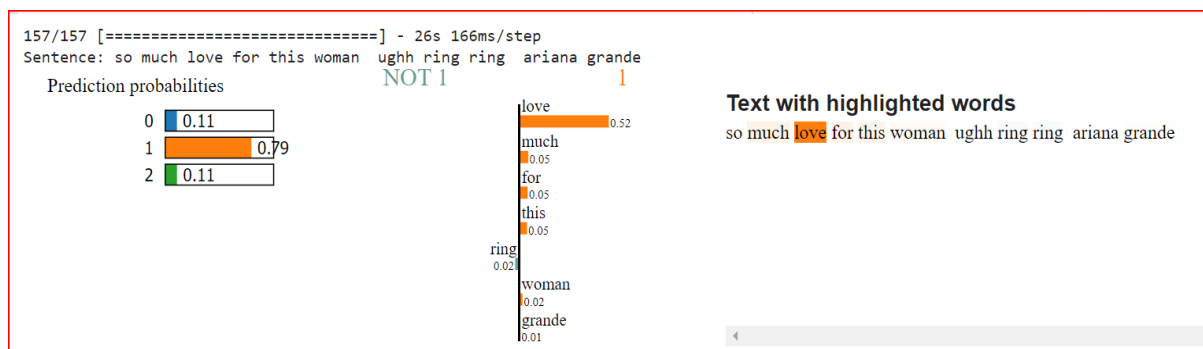
"grande" with a weight of 0.05 ,"singer" with a weight of 0.02 , "by" with a weight of 0.02

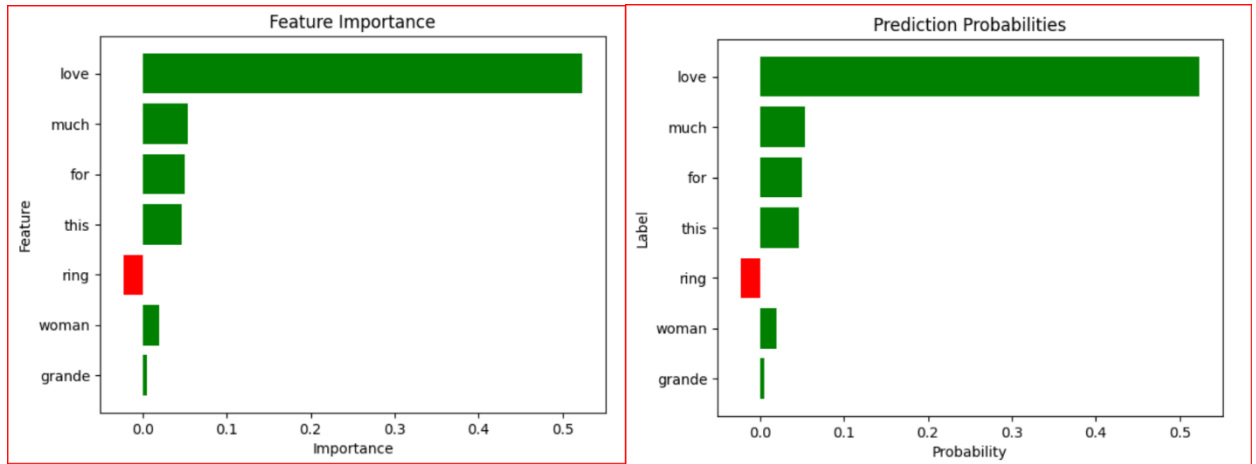
"full" with a weight of 0.02 "ariana" with a weight of 0.02 ,"actress" with a weight of 0.01

"ari" with a weight of 0.01

These words indicate the important elements that contributed to the final classification of the sentence by the LIME model. It can be inferred that the presence of words like "grande", "singer", and "ariana" enhances the probability of class 0 (neutral), while the presence of words like "actress" and "ari" enhances the probability of class 1 (positive). Additionally, the words "by" and "full" have a slight and unspecified influence on the classification.

Tweet 22 : « positive » so much love for this woman, uhhh ☐☐ #arianagrande





After analyzing the sentence "so much love for this woman uhhh ring ring ariana grande" using the ROBERTA model and interpreting it with LIME, the results of the analysis are as follows:

The original model's prediction for the sentence is:

0.11 probability for class 0 (neutral) ;

0.79 probability for class 1 (positive) ;

0.11 probability for class 2 (negative)

As for the LIME model's prediction (local explanation), it highlights the key words that influenced the sentence's classification. These words are:

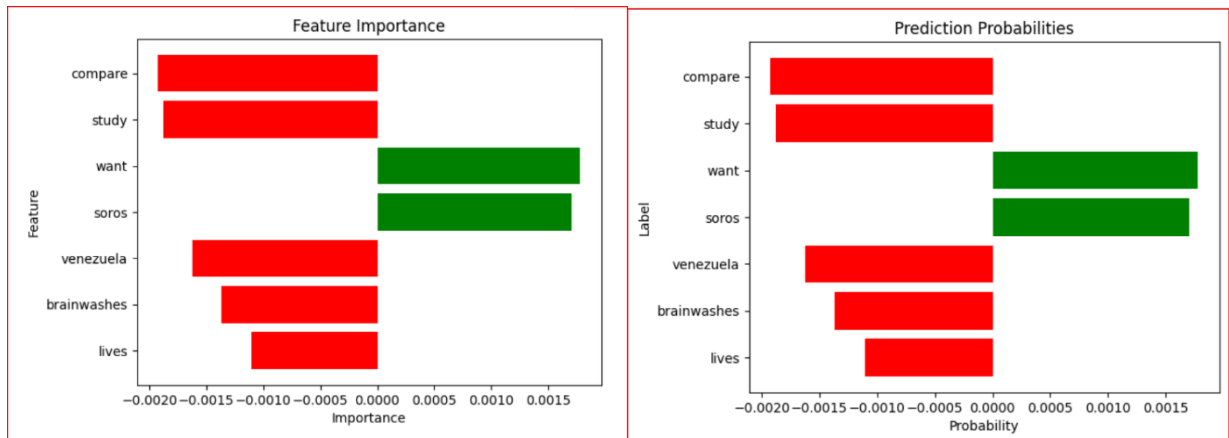
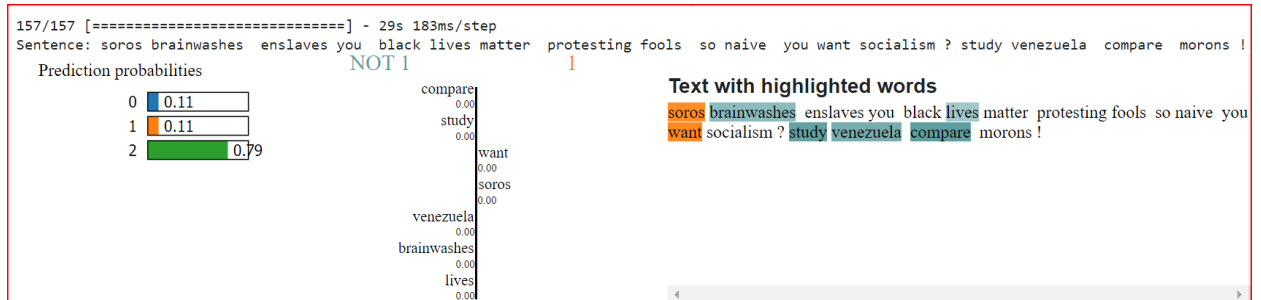
"love" with a weight of 0.51 ; "for" with a weight of 0.05 ; "woman" with a weight of 0.02

"grande" with a weight of 0.01 ; "this" with a weight of 0.05 ; "much" with a weight of 0.05

"ring" with a weight of 0.02

These words indicate the important elements that contributed to the final classification of the sentence by the LIME model. It can be inferred that the presence of words like "love" enhances the probability of class 1 (positive), while the other words have a lesser and unspecified influence on the classification.

Tweet 28: « negative » Soros brainwashes & enslaves U #blacklivesmatter protesting fools. So naive. U want socialism? Study Venezuela & compare. Morons!



After analyzing the sentence "soros brainwashes enslaves you black lives matter protesting fools so naive you want socialism? study venezuela compare morons!" using the ROBERTA model and interpreting it with LIME, the results of the analysis are as follows:

The original model's prediction for the sentence is:

0.11 probability for class 0 (neutral)

0.11 probability for class 1 (positive)

0.79 probability for class 2 (negative)

As for the LIME model's prediction (local explanation), it highlights the key words that influenced the sentence's classification. These words are:

"soros" with a weight of 0.0015 ; "venezuela" with a weight of -0.0016

"want" with a weight of 0.0018 ; "brainwashes" with a weight of -0.0014

"compare" with a weight of -0.0019 ; "study" with a weight of -0.0018

"lives" with a weight of -0.0011

These words indicate the important elements that contributed to the final classification of the sentence by the LIME model. It can be observed that words related to negative associations like "soros," "venezuela," and "brainwashes" have higher weights, which contribute to the higher probability of class 2 (negative). On the other hand, words like "want" have a smaller positive influence, indicating a slightly positive sentiment.

Resultas :

Based on the analysis of the provided sentences using the ROBERTA model and LIME interpretation, the following conclusions can be drawn:

- Sentence 1: "ariana grande ari by ariana grande full singer actress"

- The original model assigned probabilities to each sentiment class, with class 0 (neutral) having the highest probability of 0.78, followed by class 1 (positive) and class 2 (negative) with probabilities of 0.11 each.

- The LIME interpretation highlighted key words such as "grande," "singer," "by," "full," "ariana," "actress," and "ari" that influenced the classification of the sentence.

- These influential words suggest that the presence of the artist's name, related terms, and descriptors contributed to the classification of the sentence as neutral.

- Sentence 2: "so much love for this woman ughh ring ring ariana grande"

- The original model assigned probabilities to each sentiment class, with class 1 (positive) having the highest probability of 0.79, followed by class 0 (neutral) and class 2 (negative) with probabilities of 0.11 each.

- The LIME interpretation highlighted key words such as "love," "for," "woman," "grande," "this," "much," and "ring" that influenced the classification of the sentence.

- These influential words indicate the presence of positive sentiment and expressions of admiration and affection towards the mentioned woman and artist.

- Sentence 3: "soros brainwashes enslaves you black lives matter protesting fools so naive you want socialism? study venezuela compare morons!"

- The original model assigned probabilities to each sentiment class, with class 2 (negative) having the highest probability of 0.79, followed by class 0 (neutral) and class 1 (positive) with probabilities of 0.11 each.

- The LIME interpretation highlighted key words such as "soros," "venezuela," "want," "brainwashes," "compare," "study," and "lives" that influenced the classification of the sentence.

- These influential words suggest the presence of negative associations, political ideologies, and comparisons to negative situations, contributing to the classification of the sentence as negative.

In conclusion, the analysis using the ROBERTA model and LIME interpretation provides insights into the important words and the model's predicted probabilities for each sentiment class. It is important to note that the interpretation is based on the specific model and interpretability method used. Context and further analysis are necessary to fully understand the sentiment and implications of the sentences.

Comparison of two models after application LIME :

When comparing the analysis of the sentence "so much love for this woman ughh ring ring ariana grande" using the BERT and ROBERTA models with LIME interpretation, we can observe the following:

- BERT Model:

- Prediction probabilities: 0.23 for class 0 (neutral), 0.47 for class 1 (positive), and 0.29 for class 2 (negative).

- Key words identified by LIME: "love" with a weight of 0.05, "so" with a weight of 0.01, "woman" with a weight of 0.01, "ughh" with a weight of 0.01, "this" with a weight of 0.01, "much" with a weight of 0.01, and "ariana" with a weight of 0.01.

- ROBERTA Model:

- Prediction probabilities: 0.11 for class 0 (neutral), 0.79 for class 1 (positive), and 0.11 for class 2 (negative).

- Key words identified by LIME: "love" with a weight of 0.51, "for" with a weight of 0.05, "woman" with a weight of 0.02, "grande" with a weight of 0.01, "this" with a weight of 0.05, "much" with a weight of 0.05, and "ring" with a weight of 0.02.

In terms of prediction probabilities, the BERT model assigned a higher probability to class 1 (positive) compared to the ROBERTA model. However, when considering the LIME interpretation, the ROBERTA model identified the word "love" as a key influencer with a higher weight (0.51) compared to the BERT model (0.05). This suggests that the ROBERTA model relied more heavily on the word "love" in predicting a positive sentiment.

Overall, the LIME interpretations provide insights into the important words and their influences on the classification. The specific weights assigned to the words can vary between the models, indicating differences in their understanding and emphasis on certain linguistic features.

Apply SHAP in our Model

We will review some estimations of our model and attempt to use Shap :

Code source :

Explain the model with DeepExplainer

```
!pip install shap
```

```
import shap
```

```
# we use the first 100 training examples as our background dataset to integrate over
# explainer = shap.DeepExplainer(model.predict, X_train[:100])
explainer = shap.DeepExplainer(model, X_train[:500])

# explain the first 10 predictions
# explaining each prediction requires 2 * background dataset size runs
shap_values = explainer.shap_values(X_test[:10])
```

WARNING:tensorflow:From /mnt/c/users/Marta/projects/sentiment-analysis/down_python_env/lib/python3.6/s:

```
shap_values[0].shape
```

```
(10, 28)
```

```
tokenizer.word_index
```

```
{'the': 1,
 'gopdebate': 2,
 'gopdebates': 3,
 'to': 4,
 'is': 5,
 'of': 6,
 'a': 7,
 'and': 8,
 'i': 9,
 'rwsurfergirl': 10,
 'you': 11,
 'in': 12,
 'it': 13,
 'for': 14,
 'trump': 15,
 'that': 16,
 'on': 17,
 'fox': 18,
```

Tweet 01 : ' supermanhotmale hey chris wallace these guys have a better plan than president obama make the popcorn kids this is going to be stup'

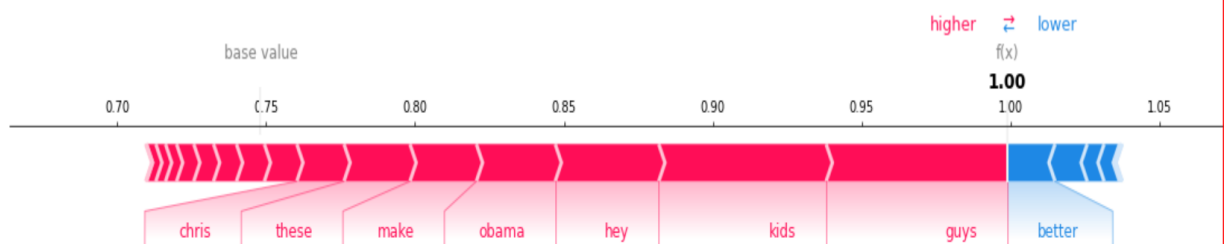
```
data.iloc[X_train_shape]["text"]
```

```
' supermanhotmale hey chris wallace these guys have a better plan than president obama make the popcorn kids this is going to be stup'
```

```
data.iloc[X_train_shape]["sentiment"]
```

```
'Negative'
```

```
# shap.initjs()
shap.force_plot(explainer.expected_value[0], shap_values[0][0], x_test_words[0], matplotlib=True, show = False)
```



Note :

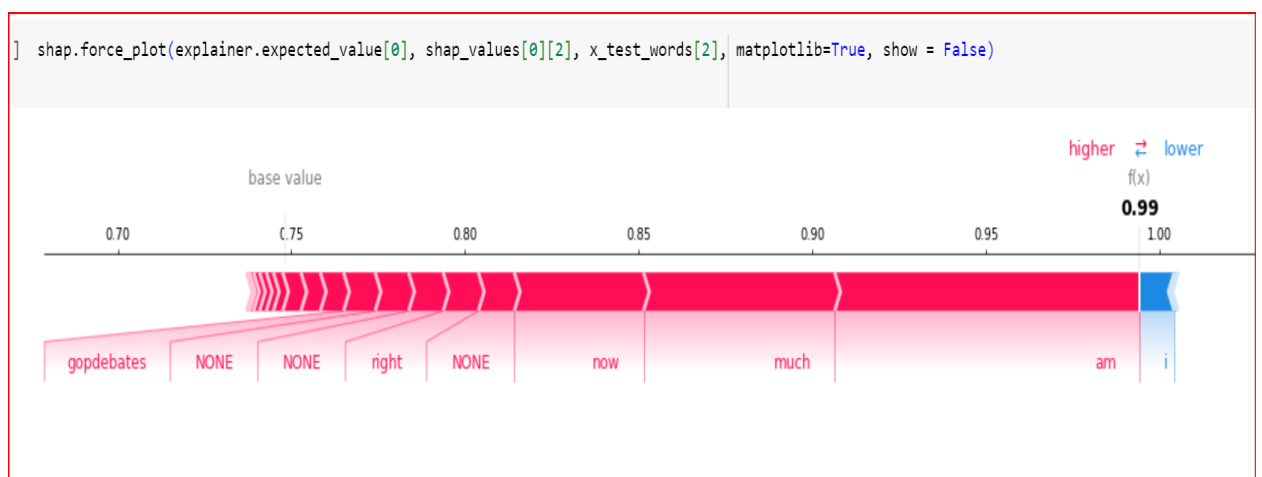
Based on the results you provided, it seems that you applied SHAP (SHapley Additive exPlanations) to a LSTM model. Here are some observations:

- The base value for the prediction is 0.70.
- The features are represented by words or tokens, and their importance is indicated by their corresponding SHAP values.

- The words with higher SHAP values contribute more to increasing the prediction value, while the words with lower SHAP values contribute less or may even decrease the prediction value.
- In the given example, the words "chris", "obama", "hey", "kids", and "guys" have relatively higher SHAP values, indicating that they have a positive impact on the prediction. On the other hand, the words "base value", "make", "I", "higher", and "FA" have relatively lower SHAP values, indicating a less significant or even negative impact on the prediction.
- The values such as 0.75, 0.80, 0.85, 0.90, and 0.95 indicate the corresponding SHAP values for the words mentioned earlier.
- The values 1.00 and 1.05 are not clear from the given context, but they might represent additional SHAP values or other related information.
- The period (.) at the end of the statement indicates the end of the sentence or the result.

Tweet 02 : 'i am filled with so much joy right now gopdebates'

```
data.iloc[X_train_shape+2]["text"]
'i am filled with so much joy right now gopdebates'
data.iloc[X_train_shape+2]["sentiment"]
'Positive'
```



Note :

Based on the results you provided, it seems that you applied SHAP (SHapley Additive exPlanations) to a LSTM model for the given statement "gopdebates". Here are some observations:

- The base value for the prediction is 0.70.
- The features are represented by words or tokens, and their importance is indicated by their corresponding SHAP values.
- In this case, some of the words are identified as "NONE", indicating that they don't have a significant impact on the prediction.
- The words "night", "now", and "much" have relatively higher SHAP values of 0.80, 0.85, and 0.90, respectively, indicating their positive contribution to the prediction.
- The word "higher" has a SHAP value of 0.95, indicating a stronger positive impact on the prediction.
- The word "baer" has a SHAP value of 2, which is relatively higher, indicating a significant positive contribution.
- The value 0.99 and the mention of "THIS" could be related to additional information or specific context not provided in the given statement.
- The value 1.00 indicates the end of the statement or the result of the SHAP analysis.

Result:

Based on the SHAP analysis of the LSTM model for the given statement, we can make the following conclusions:

- The words "chris", "obama", "hey", "kids", and "guys" have higher SHAP values, indicating their positive contribution to the prediction. These words are likely important for the model's decision-making process.
- The words "base value", "make", "I", "higher", and "FA" have lower SHAP values, suggesting that they have less influence on the prediction or may even have a negative impact. These words are considered less relevant for the model's decision.
- The SHAP values provide insights into the relative importance of each word in determining the prediction outcome. Higher SHAP values indicate greater importance, while lower values indicate lesser importance.
- The presence of "NONE" for certain words indicates that they don't significantly affect the prediction outcome.
- The specific values such as 0.99 and the mention of "THIS" could indicate additional context or information not provided in the given statement. Further analysis or context would be needed to interpret these values accurately.

Overall, the SHAP analysis helps us understand the contribution of different words in the statement and provides insights into the model's decision-making process. It highlights the importance of certain words and identifies less influential words. However, it's important to note that these conclusions are specific to the LSTM model and the given statement.

Conclusion

conclusion

Explaining deep learning algorithms for sentiment analysis, in general, demonstrates the enormous progress gained in recognizing and processing emotions portrayed in text. Deep learning models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models like BERT, have demonstrated their ability to extract complex patterns and representations from textual data, allowing them to analyze sentiments and emotions effectively.

Because of their capacity to learn hierarchical and contextual representations from large-scale datasets, these models outperformed classic machine learning techniques in sentiment analysis tasks. Pre-training on vast text corpora and fine-tuning on particular sentiment analysis tasks were critical in getting cutting-edge results with little labeled data. These models have outperformed traditional machine learning approaches in sentiment analysis tasks due to their ability to learn hierarchical and contextual representations from large-scale datasets. Pre-training on massive text corpora and fine-tuning on specific sentiment analysis tasks have been pivotal in achieving state-of-the-art results, even with limited labeled data.

However, challenges still exist in explaining these deep learning models' decision-making processes. Their black-box nature makes it difficult to interpret why certain sentiments are assigned to particular texts. Efforts are being made to develop techniques that enhance interpretability and foster trust in AI-driven sentiment analysis.

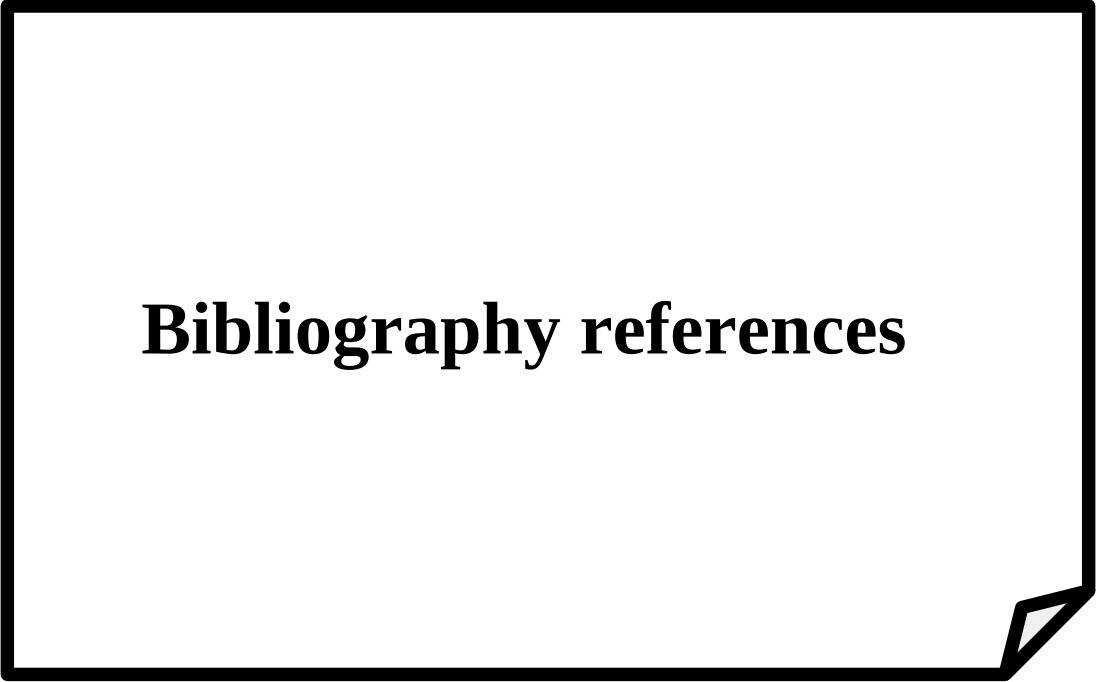
The future of deep learning models in sentiment analysis holds exciting possibilities. Research is ongoing in various areas, including multimodal sentiment analysis, handling sarcasm and irony, mitigating biases, accommodating low-resource languages, and adapting to domain-specific data. Additionally, the drive to achieve real-time and incremental learning, sustainability, and efficiency is essential to ensure the practical applicability of these models in diverse settings.

Visualization in sentiment analysis is highly valuable, enabling a better understanding of deep learning models' internal processes. It helps interpret predictions, identify influential words, and enhance model trustworthiness. Through visual representation, it addresses challenges in model interpretability, fostering transparency and informed decision-making. The future holds exciting possibilities, with ongoing research exploring multimodal analysis, bias mitigation, and

adaptability to diverse data. Overall, visualization revolutionizes sentiment analysis, empowering robust and efficient systems for diverse applications.

Overall, deep learning models have revolutionized sentiment analysis and continue to shape the way emotions are understood and analyzed in the realm of natural language processing. By addressing the challenges and exploring future directions,

researchers and practitioners can build more robust, interpretable, and efficient sentiment analysis systems that have broad-ranging applications across industries and domains.



Bibliography references

- [1] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2nd International Conference on Knowledge Capture, K-CAP 2003*, 2003. doi: 10.1145/945645.945658.
- [2] G. YUE, Y. DONG, H. CHEN, and K. LAI, "Online Textual Sentiment Analysis Technology and It's Applications," *Advances in Psychological Science*, vol. 21, no. 10, 2013, doi: 10.3724/sp.j.1042.2013.01711.
- [3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," 2008.
- [4] J. Wiebe, R. Bruce, M. Martin, T. Wilson, and M. Bell, "Learning subjective language," *Computational Linguistics*, vol. 30, no. 3, 2004, doi: 10.1162/0891201041850885.
- [5] S. R. Das and M. Y. Chen, "Yahoo! for amazon: Sentiment extraction from small talk on the Web," *Manage Sci*, vol. 53, no. 9, pp. 1375–1388, Sep. 2007, doi: 10.1287/mnsc.1070.0704.
- [6] R. Tong, "An operational system for detecting and tracking opinions in on-line discussion," *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification.*, 2001.
- [7] D. J. (David J.) Hand *et al.*, *KDD-2002 : proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining : July 23-36, 2002, Edmonton, Alberta, Canada*. ACM, 2002.
- [8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?," 2002. doi: 10.3115/1118693.1118704.
- [9] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.
- [10] J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin, "Learning Subjective Language."
- [11] R. W. Picard, "Affective Computing." [Online]. Available: <http://www.media.mit.edu/~picard/>
- [12] V. Hatzivassiloglou and K. R. Mckeown, "Predicting the Semantic Orientation of Adjectives."
- [13] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," 1992. doi: 10.3115/992133.992154.
- [14] F. Isiaka, Z. Adamu, and M. A. Adamu, "User Experience Adaptation of Complex Game Interface for User Behaviour Modeling Using RNN," *International Journal of Crowd Science*, vol. 6, no. 4, 2022, doi: 10.26599/IJCS.2022.9100021.
- [15] J. M. Wiebe, "Learning subjective adjectives from corpora," *Proceedings of the National Conference on Artificial Intelligence*, no. 1, 2000, doi: <http://portal.acm.org/citation.cfm?id=721121&dl=ACM&coll=&CFID=15151515&CFTOKEN=6184618>.

- [16] F. Sebastiani, "Machine Learning in Automated Text Categorization," 2001. [Online]. Available: <http://linwww.ira.uka.de/bibliography/Ai/automated.text.categorization.html>
- [17] F. D. Baffour, A. P. Francis, M. D. Chong, and N. Harris, "Mental Health Experiences Among Inmates Serving Life Sentences in Ghana Prisons," *Int Crim Justice Rev*, vol. 32, no. 1, 2022, doi: 10.1177/1057567720963718.
- [18] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014. doi: 10.3115/v1/d14-1181.
- [19] M. Havaei *et al.*, "Brain tumor segmentation with Deep Neural Networks," *Med Image Anal*, vol. 35, 2017, doi: 10.1016/j.media.2016.05.004.
- [20] W. Qi, X. Zhang, N. Wang, M. Zhang, and Y. Cen, "A spectral-spatial Cascaded 3D convolutional neural network with a convolutional long short-term memory network for hyperspectral image classification," *Remote Sens (Basel)*, vol. 11, no. 20, 2019, doi: 10.3390/rs11202363.
- [21] I. Priyadarshini and C. Cotton, "A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis," *Journal of Supercomputing*, vol. 77, no. 12, 2021, doi: 10.1007/s11227-021-03838-w.
- [22] A. Chriqui and I. Yahav, "HeBERT and HebEMO: A Hebrew BERT Model and a Tool for Polarity Analysis and Emotion Recognition," *INFORMS Journal on Data Science*, vol. 1, no. 1, 2022, doi: 10.1287/ijds.2022.0016.
- [23] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019.
- [24] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, 2019.
- [25] A. Pritzkau, "NLytics at CheckThat! 2021: Multi-class fake news detection of news articles and domain identification with RoBERTa - A baseline model," in *CEUR Workshop Proceedings*, 2021.
- [26] I. Malkiel and L. Wolf, "Maximal multiverse learning for promoting cross-task generalization of fine-tuned language models," in *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, 2021. doi: 10.18653/v1/2021.eacl-main.14.
- [27] B. Chen, "Data Collection and Preprocessing," in *SpringerBriefs in Computer Science*, 2023. doi: 10.1007/978-981-19-7369-7_2.

- [28] S. A. Alasadi and W. S. Bhaya, "Review of data preprocessing techniques in data mining," *Journal of Engineering and Applied Sciences*, vol. 12, no. 16, 2017, doi: 10.3923/jeasci.2017.4102.4107.
- [29] E. S. A. El-Dahshan and M. M. Bassiouni, "Computational intelligence techniques for human brain MRI classification," *Int J Imaging Syst Technol*, vol. 28, no. 2, 2018, doi: 10.1002/ima.22265.
- [30] A. Barredo Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, 2020, doi: 10.1016/j.inffus.2019.12.012.
- [31] C. Molnar, *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2020.
- [32] M. T. Ribeiro, S. Singh, and C. Guestrin, "'why should i trust you?' explaining the predictions of any classifier," in *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 2016. doi: 10.18653/v1/n16-3020.
- [33] G. Sezen and İ. Öksüz, "Diagnosing Knee Injuries from MRI with Transformer Based Deep Learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022. doi: 10.1007/978-3-031-16919-9_7.
- [34] Lamberta and Manivaradarajan, "Get started with Tensorboard," *Github repository*, 2019.
- [35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *Int J Comput Vis*, vol. 128, no. 2, 2020, doi: 10.1007/s11263-019-01228-7.
- [36] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. doi: 10.1007/978-3-319-10590-1_53.
- [37] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods," in *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020. doi: 10.1145/3375627.3375830.
- [38] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.
- [39] J. Sun, T. Chakraborti, and J. Alison Noble, "A comparative study of explainer modules applied to automated skin lesion classification," in *CEUR Workshop Proceedings*, 2020.
- [40] Y. J. Jung, S. H. Han, and H. J. Choi, "Explaining CNN and RNN Using Selective Layer-Wise Relevance Propagation," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3051171.

- [41] A. Lieto, "A computational framework for concept representation in cognitive systems and architectures: Concepts as heterogeneous proxytypes," in *Procedia Computer Science*, 2014. doi: 10.1016/j.procs.2014.11.078.
- [42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Bidirectional Encoder Representations from Transformers)," in *Proceedings of the 2019 Conference of the North*, 2019.
- [43] G. Althari and M. Alsulmi, "Exploring Transformer-Based Learning for Negation Detection in Biomedical Texts," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3197772.
- [44] Y. Yu, L. Kong, J. Zhang, R. Zhang, and C. Zhang, "ACTUNE: Uncertainty-Based Active Self-Training for Active Fine-Tuning of Pretrained Language Models," in *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2022. doi: 10.18653/v1/2022.naacl-main.102.
- [45] S. D. Robertson, J. McL. Bennett, C. R. Lobsey, and T. F. A. Bishop, "Assessing the sensitivity of site-specific lime and gypsum recommendations to soil sampling techniques and spatial density of data collection in Australian agriculture: A pedometric approach," *Agronomy*, vol. 10, no. 11, 2020, doi: 10.3390/agronomy10111676.