SCHOOL OF INNOVATIVE TECHNOLOGIES AND ENGINEERING

INDUSTRIAL SYSTEMS ENGINEERING

&

POLYTECHNICS MAURITIUS LTD (PML)

Interim Report

Prisca Venkiah

Supervisor: Mr. Zoubeir Aungnoo

Table of Contents

Introduction

Brand reputation is a crucial aspect of modern business, and maintaining a positive image is vital for success. In recent years, sentiment analysis has emerged as a powerful tool for monitoring and managing brand reputation. Sentiment analysis software uses natural language processing techniques to automatically identify and extract subjective information from text, such as social media posts, reviews, and news articles. This literature review aims to explore the current state of research on sentiment analysis for brand reputation management.

Literature Review

The field of sentiment analysis has seen significant advancement over the years, driven by both necessity and technological progress. As the volume of user-generated content on the internet has grown exponentially, so too has the need for efficient and accurate sentiment analysis systems. Traditional approaches, such as keyword-based methods or machine learning techniques using hand-crafted features, have been limited in their ability to fully capture the nuances and complexities of human sentiment.

Sentiment analysis, also known as opinion mining, is a field within NLP that builds systems to identify and extract subjective information from text. Early research on sentiment analysis primarily focused on classifying text as positive, negative, or neutral. Pang and Lee (2002) were among the pioneers in this field, employing machine learning techniques for sentiment classification of movie reviews. However, traditional machine learning approaches required extensive manual feature engineering and often struggled with the subtleties of human language.

More recent research has shifted towards deep learning-based methods, which demonstrate superior performance by learning features automatically from large amounts of data. However, despite their effectiveness, these methods still grapple with the inherent complexity and ambiguity of human language. Sentiments are often context-dependent, and their accurate detection requires a thorough understanding of language semantics, syntax, and pragmatics. Sarcasm, irony, and domain-specific jargon present additional challenges.

The advent of deep learning brought significant advancements in sentiment analysis. Researchers began to use models like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) that could learn features automatically from data. For instance, Socher et al.

(2013) proposed a Recursive Neural Tensor Network to model sentence structure and semantics for sentiment classification.

Despite these advancements, deep learning models still faced challenges in handling long-term dependencies and understanding the context of words in a sentence. Attention mechanisms were introduced to address these issues by allowing models to focus on the most relevant parts of the input for making predictions (Bahdanau et al., 2014).

Review of Models/Concept

Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting, or spoken words. Unlike traditional neural networks, RNNs retain a form of memory as they process sequences of inputs. This characteristic makes them particularly suited to tasks like unsegmented, connected handwriting recognition or speech recognition.

Structure of RNN

The fundamental feature of RNNs is their hidden state, which captures information about a sequence. An RNN takes as input a sequence of vectors (x1, x2, ..., xt). At each step t, it combines the current input vector xt and the previous hidden state ht-1 to produce a new hidden state ht. This process gives RNNs the ability to build up memory of what has been seen so far in the sequence.

However, standard RNNs suffer from a few significant issues. One of them is the problem of "vanishing gradients" and "exploding gradients." These problems occur during the backpropagation process when the network is learning from the errors in its predictions. For very long sequences, the gradients – which are used to update the weights of the network – can become very small (vanish) or very large (explode). This makes the learning process unstable and inefficient.

Another issue is that standard RNNs typically have a "short-term memory." They are good at handling sequences where the relevant information is close together, but they struggle when the sequences are long and the relevant information is spread out. This is because the influence of a given input on the hidden state, and hence on the network's outputs, tends to disappear over time.

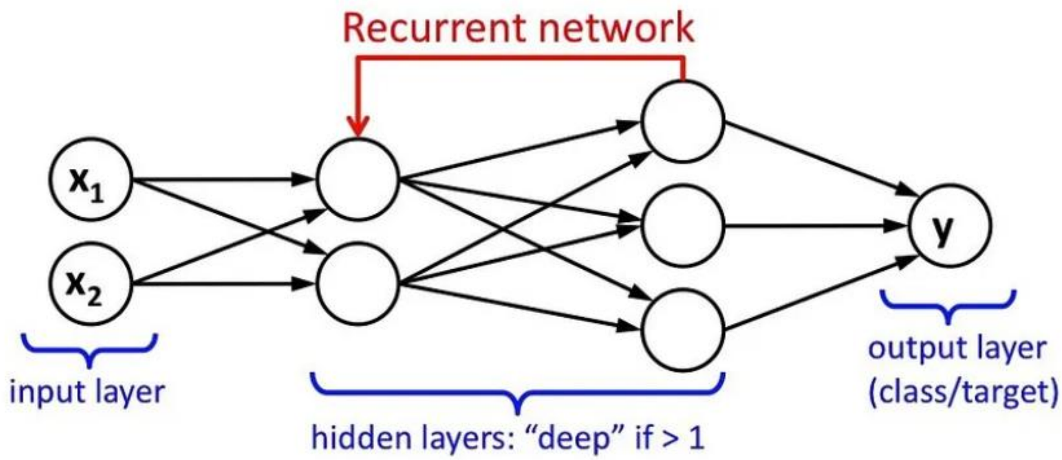

Figure 1: Recurrent Neural Network Architecture

Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a type of deep learning model traditionally used in image processing tasks, due to their ability to recognize local patterns in an image by convolving small windows, or "filters", over the input data. These filters allow a CNN to capture spatial and temporal dependencies in the data, which are essential for image understanding.

However, CNNs have also been applied to the domain of natural language processing, including sentiment analysis. In the context of sentiment analysis, a CNN can take a sequence of word embeddings (vector representations of words) as input and apply convolutional filters to extract higher-level features that capture local semantic information.

For example, in a simple sentiment analysis scenario, a 1D convolution could be applied to a sequence of word embeddings to capture bi-grams or tri-grams (combinations of 2 or 3 words) that often carry sentiment information, such as "not good" or "very happy". These local features are then pooled (e.g., via max pooling) to create a fixed-size vector representation of the text, which can be used for sentiment classification.

However, while CNNs can capture local semantic features in text data, they are not without their limitations. One of the main challenges of using CNNs for sentiment analysis is their inability to capture long-term dependencies in the text data.

In sentiment analysis, the sentiment of a piece of text is often determined not just by individual words or bi-grams/tri-grams, but by the overall context of the text, which can span long sequences of words. While CNNs can capture local dependencies within the range of their filters, they struggle to capture relationships between words or phrases that are further apart.

Moreover, CNNs treat each feature extracted from the text independently and do not model the interactions between these features. This can be problematic for sentiment analysis, where the sentiment can be influenced by the interplay of different words and phrases.

Finally, another limitation is that standard CNNs do not take into account the order of the words in the text, which is crucial for understanding the sentiment. For example, the phrases "this is good, not bad" and "this is bad, not good" have very different sentiments, but would be treated the same by a standard CNN.

Due to these limitations, CNNs are often not the first choice for sentiment analysis tasks. Recurrent Neural Networks (RNNs) and Transformer-based models, which are better suited to capturing long-term dependencies and the order of words in text, are typically preferred.

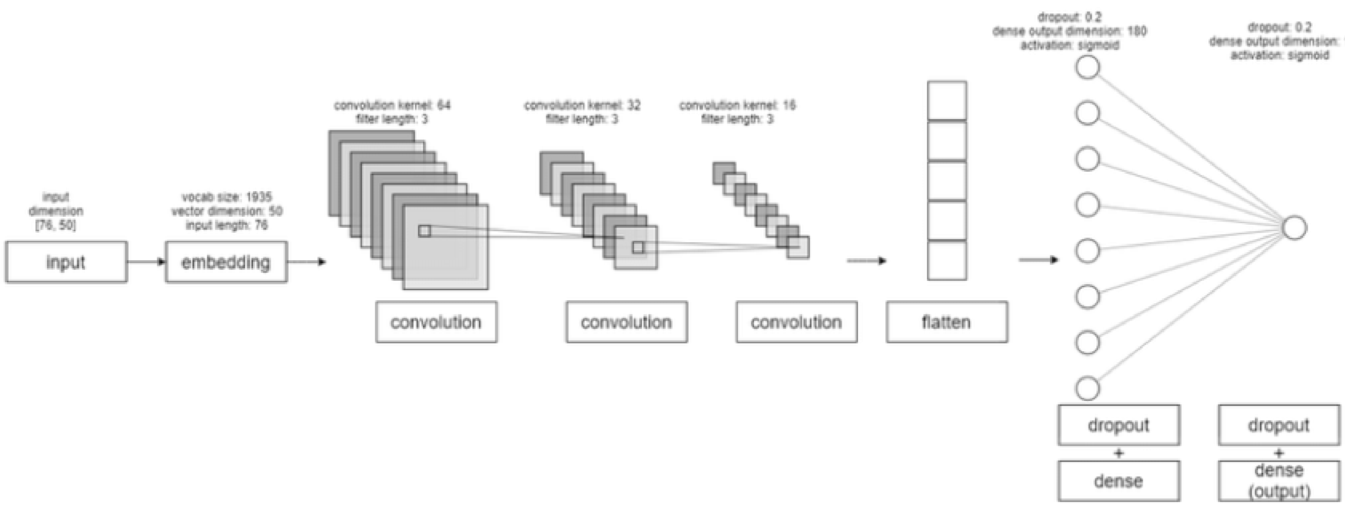

Figure 2: Convolutional Neural Network Architecture

DistilBERT Model

DistilBERT is a smaller, faster, and more efficient version of the BERT model, specifically designed for tasks requiring lower memory usage and faster inference times (Sanh et al., 2019). It retains around 95% of BERT's performance while using 40% fewer parameters and running 60% faster. DistilBERT's architecture is based on the transformer architecture (Vaswani et al., 2017), with self-attention mechanisms that enable the model to understand the contextual relationships between words in a sequence.

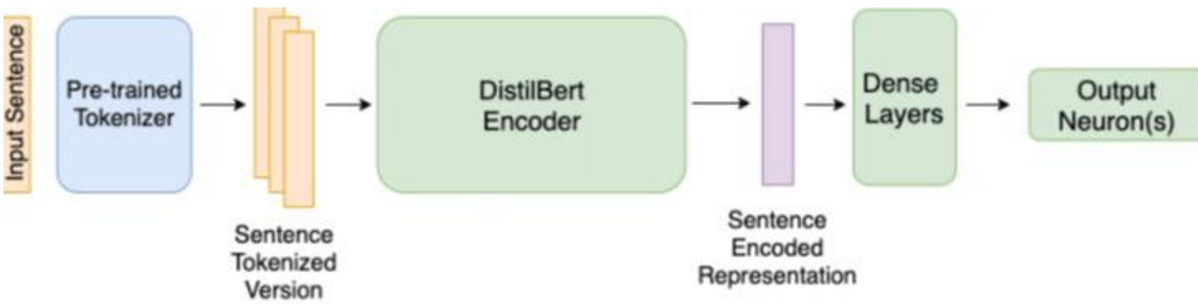

Figure 3: DistilBERT Architecture

Pretraining and Fine-tuning

Like BERT, DistilBERT is pretrained on a large corpus of text using unsupervised learning. The pretraining involves two tasks: masked language modeling (MLM) and next sentence prediction (NSP). MLM encourages the model to learn contextual word representations by randomly masking a percentage of the input tokens and training the model to predict the masked tokens based on the surrounding context. NSP trains the model to predict whether two sentences are consecutive or not.

After pretraining, DistilBERT is fine-tuned on specific tasks like sentiment analysis, named-entity recognition, or question answering using supervised learning. In our program, we use a version of DistilBERT fine-tuned for the Stanford Sentiment Treebank (SST-2) dataset, which consists of movie reviews labeled as positive or negative.

Tokenization

DistilBERT uses WordPiece tokenization (Wu et al., 2016) to break text into subword units. This technique allows the model to represent rare or out-of-vocabulary words by splitting them into smaller, known subword units. The tokenization process also adds special tokens, such as [CLS] and [SEP], to facilitate classification tasks and distinguish between different input sequences. [CLS] token is used as the first token in the input sequence and [SEP] denotes the **end of a sentence**. BERT takes all the inputs as a single sequence. If we need to provide more than one input as in our case where our input will be premise and hypothesis, [SEP] token helps the model to understand input properly.

Scope of the Project

We are using DistilBERT model for this project because it is more accurate and appropriate for sentiment analysis. This project presents a sentiment analysis program that leverages the DistilBERT model for natural language understanding and Streamlit as an interactive front-end. The program processes an input Excel file containing user comments, classifies the sentiment of each comment, and computes the average sentiment score. The Project delves into the underlying concepts of the DistilBERT model, its relevance to sentiment analysis, and the integration of the model within a Streamlit web application.

Streamlit

Streamlit is an open-source Python library designed for creating interactive web applications with minimal development effort (Adrien et al., 2019). It is particularly well-suited for data-centric applications that require user interaction with data visualizations, tables, or charts. Streamlit apps consist of Python scripts that use built-in functions to define user interface components like file uploaders, sliders, and data tables.

Sentiment Analysis Program

The presented program employs the Hugging Face Transformers library to load the DistilBERT model and tokenizer. We create a sentiment analysis pipeline, which tokenizes the input text, feeds it through the DistilBERT model, and outputs the sentiment prediction.

The program accepts an Excel file containing a column of comments. It then uses the sentiment analysis pipeline to classify each comment as positive or negative and calculates a sentiment score. The results are displayed in a table, along with the average sentiment score.

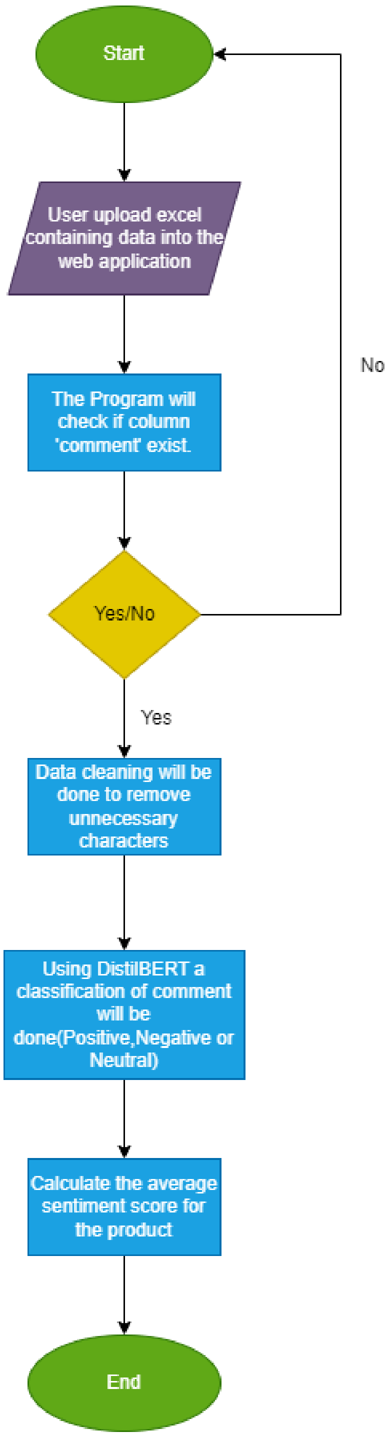

Figure 4: Flow Chart

Performance and Limitations

In this section, we discuss the performance and limitations of the presented sentiment analysis program.

Performance

The DistilBERT model demonstrates high performance in sentiment analysis tasks, largely due to its transformer-based architecture and fine-tuning on the SST-2 dataset. The model's ability to capture context-dependent meaning and relationships between words makes it well-suited for understanding the nuances of sentiment in text. Moreover, the use of subword tokenization allows the model to handle out-of-vocabulary words effectively, leading to better generalization on unseen text.

Limitations

Despite its effectiveness, the DistilBERT model still has some limitations. For instance, it may struggle to capture the sentiment in text with complex language constructs, sarcasm, or domain-specific jargon. Additionally, the model is designed for English-language text, which may limit its applicability to texts in other languages.

Another limitation is the program's reliance on a single column named "comment" in the input Excel file. This constraint may require users to preprocess their data before using the program. Furthermore, the program currently supports binary sentiment classification, which may not suffice for more granular sentiment analysis tasks, such as classifying sentiment on a five-point scale or identifying different emotions.

Aims & Objectives

There exist situations where companies are not aware of their own products limitation. Thus, their competitors tend to capitalize on those limitation to gain a higher market share. There are times, where the company is not even aware that a product is performing poorly in the market. So how do we improve this? This is where, Sentiment analysis comes into play. The solution for this problem would be either to start surveys which tend to have a high probability of false positive results and on top of that, it is costly and time-consuming. The second option would be to process the reviews/comments of customers and coming up with a sort of sentiment score and a recommender for the product.

Sentiment analysis is an essential task in natural language processing, with applications ranging from social media monitoring to customer feedback analysis. The advent of transformer-based models has significantly improved the performance of sentiment analysis systems. In this project, we present a sentiment analysis program that employs the DistilBERT model, a lighter and faster variant of the BERT architecture, to analyze comments from an Excel file. We also use the Streamlit framework to create an interactive web-based interface for the program.

Conclusion

In this project, we presented a sentiment analysis program that uses the DistilBERT model to classify comments in an Excel file as positive or negative. The program leverages the high performance of the DistilBERT model and the simplicity of the Streamlit framework to provide an interactive and user-friendly web-based interface.

While the program is effective for binary sentiment classification tasks, it is important to consider its limitations when dealing with complex language constructs, domain-specific jargon, or texts in languages other than English. Future work could involve extending the program to support more granular sentiment classification, emotion recognition, or multilingual analysis.

Related Literature:

Liu et al. (2018) investigated the impact of social media sentiment on brand reputation using machine learning techniques. The authors proposed a hybrid approach that combines a lexicon-based approach with a machine learning algorithm to improve the accuracy of sentiment classification. The study found that sentiment analysis can be an effective tool for monitoring brand reputation and identifying potential risks.

Jha et al. (2019) conducted a study on the impact of online reviews on brand reputation. The authors used sentiment analysis to classify reviews as positive, negative, or neutral and analyzed the correlation between review sentiment and brand reputation. The study found that negative reviews can have a significant impact on brand reputation, and sentiment analysis can help identify and address these issues.

Zhang et al. (2020) proposed a sentiment analysis framework for brand reputation management based on deep learning techniques. The authors used a convolutional neural network (CNN) to classify sentiment in social media posts and news articles. The study found that the proposed framework outperforms traditional sentiment analysis methods and can be used to monitor brand reputation in real-time.

Wang et al. (2021) developed a sentiment analysis system for brand reputation management using natural language processing techniques. The authors used a rule-based approach to identify sentiment in social media posts and news articles and developed a visualization tool to help managers monitor brand sentiment over time. The study found that the system can help identify potential issues and improve brand reputation management.

Vidyashree et al. (2023) discusses the importance of sentiment analysis in gathering opinions from text data like product reviews and movie reviews, and its usefulness in various fields such as marketing and political campaigns. The focus of the paper is on sentiment analysis using Twitter data, which is a new area of study. The proposed model uses stochastic gradient descent algorithm with stochastic gradient neural network to classify tweets into positive, neural, and negative categories. The article also compares the proposed model’s performance with an existing model based on deep neural network and Forest- Whale Optimization Algorithm and concludes that the proposed model provides better results.

References

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Klingner, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.