Chapter

A Comparative Performance Evaluation of Algorithms for the Analysis and Recognition of Emotional Content

Konstantinos Kyritsis, Nikolaos Spatiotis, Isidoros Perikos and Michael Paraskevas

Abstract

Sentiment Analysis is highly valuable in Natural Language Processing (NLP) across domains, processing and evaluating sentiment in text for emotional understanding. This technology has diverse applications, including social media monitoring, brand management, market research, and customer feedback analysis. Sentiment Analysis identifies positive, negative, or neutral sentiments, providing insights into decision-making, customer experiences, and business strategies. With advanced machine learning models like Transformers, Sentiment Analysis achieves remarkable progress in sentiment classification. These models capture nuances, context, and variations for more accurate results. In the digital age, Sentiment Analysis is indispensable for businesses, organizations, and researchers, offering deep insights into opinions, sentiments, and trends. It impacts customer service, reputation management, brand perception, market research, and social impact analysis. In the following experimental research, we will examine the Zero-Shot technique on pre-trained Transformers and observe that, depending on the Model we use, we can achieve up to 83% in terms of the model's ability to distinguish between classes in this Sentiment Analysis problem.

Keywords: Sentiment Analysis, Natural Language Processing (NLP), sentiment classification, machine learning, transformers

1. Introduction

In this chapter, we present relatively new technologies in the field of sentiment analysis and examine their performance. The term "Sentiment Analysis" emerged and gained popularity around the late 2000s. While the concept of sentiment analysis had been present before, the term "Sentiment Analysis" was formally defined to refer to the automated processing and evaluation of sentiment expressed in texts, primarily in natural language texts. Since then, Sentiment Analysis has evolved and expanded with the development of advanced machine learning models, such as the scikit-learn library and later the Transformers. These powerful tools have significantly enhanced the

capabilities of sentiment analysis by providing more accurate and efficient sentiment classification algorithms. Sentiment Analysis falls into a distinct category of text classification. It involves the process of comprehending and evaluating the sentiment expressed within a sentence, paragraph, or text. The primary objective is to identify and categorize the emotional tone conveyed in these written expressions. Sentiment Analysis commonly employs various categories to capture the nuances of sentiment. Positive category encompasses texts that convey positive emotions, including pleasure, excitement, joy, optimism, and more, Negative, where this category refers to texts that express negative emotions, such as frustration, sadness, anger, worry, and others. Finally, texts falling into Neutral category do not exhibit strong positive or negative sentiments. They often maintain an impartial stance, describing information or presenting neutral viewpoints.

It is possible to expand the aforementioned categories to five by further distinguishing between "Positive" and "Negative." This can be accomplished by introducing additional subcategories: "Very Positive" and "Positive" under the Positive category, as well as "Very Negative" and "Negative" under the Negative category. With this refinement, along with the inclusion of the Neutral category, the total number of sentiment categories becomes five. However, it is essential to exercise caution when implementing such subdivisions. Introducing more categories may have implications for evaluation metrics, as it can create ambiguity between closely related terms, making it more challenging for the model to accurately differentiate and classify them.

In general, Sentiment Analysis represents a crucial area in Natural Language Processing (NLP), offering the ability to comprehend and evaluate the emotional aspects of human expressions through automated processing. By automatically analyzing and interpreting text data, Sentiment Analysis enables us to gain insights into people's sentiments, opinions, and attitudes, thereby facilitating various applications such as market research, brand monitoring, social media analysis, and customer feedback analysis.

Transformers are a class of advanced machine learning models that have emerged in recent years and have revolutionized the field of Natural Language Processing (NLP) [1]. Unlike more traditional machine algorithms, Transformers have the ability to analyze and understand complex linguistic relationships, enabling them to solve problems like Sentiment Analysis with high levels of accuracy.

On the other hand, machine learning algorithms can also be used for Sentiment Analysis, such as Naive Bayes, Decision Trees, Random Forests, Support Vector Machines, and others. These algorithms are more traditional and rely on statistical and algebraic methods. They can be successfully applied to sentence or text-level Sentiment Analysis but may not achieve the same level of accuracy and results as Transformers.

In contrast, Transformers utilize recursive neural networks and specialized models with millions of parameters, such as BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), and others, which have been trained on large volumes of text data. These models can learn rich linguistic features and word compositions to recognize and categorize sentiments with high accuracy.

In summary, while the traditional Sentiment Analysis algorithms in the scikit-learn library can produce reliable results, Transformers are more advanced models capable of handling more complex linguistic problems and achieving higher accuracy in Sentiment Analysis tasks.

In the following sections, we will dive into the Zero-Shot technique, the dataset employed, the utilization of Tokenizers in Transformers, the applications of Transformers in various tasks, and a detailed examination of four pre-trained Transformer Models. We will explore how these models function and their

experimental performance on the same dataset used in the Zero-Shot technique. Additionally, we will evaluate the effectiveness of each model based on various evaluation metrics and from an overall table of the models' metrics and a bar chart, we will see which model exhibits the best overall performance. The chapter will conclude with directions for future work.

2. Related works

In the literature, various works examine the use of transformers in sentiment analysis and in text classification. In the work presented in Prottasha et al. [2], the authors fine-tuned the BERT model, which had been pre-trained on the largest BanglaLM dataset. The model was subsequently combined with layers of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). The proposed research compared various word embedding approaches, such as Word2Vec, GloVe, fastText, and BERT. The researchers demonstrated that the transformer-based BERT model outperformed conventional techniques, achieving state-of-the-art results with sufficient fine-tuning. The study also compared several machine learning and deep learning algorithms to validate the performance of the hybrid integrated model CNN-BiLSTM (Bidirectional-LSTM). The results were analyzed using accuracy, precision, recall, F1 score, Cohen's kappa, and Receiver Operating Characteristic Area Under Curve (ROC AUC). Furthermore, the proposed model's performance was evaluated on various sentiment datasets, including the Secure Anonymised Information Linkage (SAIL) dataset, the aspect-based sentiment analysis (ABSA) dataset (cricket and restaurant parts), the BengFastText dataset, the YouTube Comments dataset, and the CogniScenti dataset. The results showed that the hybrid integrated model CNN-BiLSTM outperformed other techniques in terms of accuracy and F1 score, especially when combined with Bangla-BERT embedding.

In the work presented in Chi et al. [3], the main focus is to explore the use of pre-trained BERT models for aspect-based sentiment analysis (ABSA) tasks. The authors investigate different methods of constructing auxiliary sentences to transform ABSA into a sentence-pair classification task. These methods include question sentences, single pseudo sentences, question sentences with labels, and pseudo questions with labels. Through fine-tuning the pre-trained BERT model, they achieve new state-of-the-art results on the ABSA task using pair sentences on the datasets they evaluated. Specifically, they achieve an F1 score of 92.18 on the SentiHood dataset and an F1 score of 95.6 on the SemEval-2014 Task 4 dataset.

In the work presented in Zhang et al. [4], the authors propose a comprehensive multitask transformer network called Broad Multitask Transformer Network for Sentiment Analysis (BMT-Net) to address these issues. BMT-Net combines the strengths of feature-based and fine-tuning approaches and is specifically designed to leverage robust and contextual representations. Authors' proposed architecture ensures that the learned representations are applicable across multiple tasks through the use of multitask transformers. Furthermore, BMT-Net is capable of thoroughly learning robust contextual representations for a broad learning system, thanks to its powerful ability to explore deep and extensive feature spaces. Authors conducted experiments using two widely used datasets, namely the binary Stanford Sentiment Treebank (SST-2) and SemEval Sentiment Analysis in Twitter (Twitter). When compared to other state-of-the-art methods, authors' approach achieves superior results. Specifically, it achieves an improved F1 score of 0.778 for Twitter sentiment

analysis and an accuracy of 94.0% for the SST-2 dataset. These experimental findings not only demonstrate BMT-Net's proficiency in sentiment analysis, but also emphasize the importance of previously overlooked design choices concerning the exploration of contextual features in deep and extensive domains.

In the work presented in Junyan et al. [5], the authors propose the multimodal Sparse Phased Transformer (SPT) as a solution that mitigates the complexities associated with self-attention and memory usage. SPT employs a sampling function to generate a sparse attention matrix, effectively compressing long sequences into shorter sequences of hidden states. At each layer, SPT captures interactions between hidden states from different modalities. To further enhance the efficiency of our approach, we utilize Layer-wise parameter sharing and Factorized Co-Attention. These techniques allow for parameter sharing between Cross Attention Blocks, minimizing the impact on task performance. Authors evaluate the model using three sentiment analysis datasets and achieve comparable or superior performance compared to existing methods, all the while reducing the number of parameters by 90%. Through the experiments, authors demonstrate that SPT, along with parameter sharing, can effectively capture multimodal interactions while reducing the model size and improving sample efficiency.

In the work presented in Tan et al. [6], the authors introduce a hybrid deep learning approach that combines the benefits of both sequence models and Transformer models while mitigating the limitations of sequence models. The proposed model incorporates the Robustly optimized BERT approach and Long Short-Term Memory (LSTM) for sentiment analysis. The Robustly optimized BERT approach effectively maps words into a condensed and meaningful word embedding space, while the LSTM model excels at capturing long-range contextual semantics. Through experimental evaluations, the results demonstrate that the proposed hybrid model surpasses the performance of state-of-the-art methods. It achieves impressive F1 scores of 93, 91, and 90% on the Internet Movie Database (IMDb) dataset, Twitter US Airline Sentiment dataset, and Sentiment140 dataset, respectively. These findings highlight the effectiveness of the hybrid approach in sentiment analysis tasks.

In the work presented in Tesfagergish et al. [7], authors tackle the problem of emotion detection as a component of the broader sentiment analysis task and propose a two-stage methodology. The first stage involves an unsupervised Zero-Shot learning model, which utilizes a sentence transformer to generate probabilities for 34 different emotions. This model operates without relying on labeled data. The output of the Zero-Shot model serves as input for the second stage, which involves training a supervised machine learning classifier using ensemble learning techniques and sentiment labels. Through the proposed hybrid semi-supervised approach, authors achieve the highest accuracy of 87.3% on the English SemEval 2017 dataset. This methodology effectively combines unsupervised and supervised techniques to address sentiment analysis, incorporating emotion detection and outperforming alternative methods.

3. Zero-Shot text classification

One relatively new field in research compared to other domains is Sentiment Analysis on text datasets, where models encounter classes for the first time. These transformer models are pre-trained in natural language and utilize the Zero-Shot Text Classification technique [8].

Zero-Shot Text Classification is a machine learning technique that leverages a model's ability to classify text into categories it has never seen before. This technique is applied to texts that were not used during the model's training or were not used to develop its initial understanding of the text. This means that the model can recognize and classify data (texts) into new categories that it has not "seen" during its pre-training phase. During pre-training, these models are trained on a large volume of texts from various sources, developing a general understanding of language [9].

With this technique, the models can comprehend the meaning of the text and evaluate it in relation to predefined categories provided to them, even without having seen them before. What is important here is that they recognize the meaning of these categories. As a result, these models can classify text into new categories, increasing their flexibility and applicability in various cases, such as Zero-Shot Sentiment Analysis [10].

4. Research design and methodology

4.1 Data description

In the context of our work, we explore the "Twitter US Airline Sentiment" dataset using various variations of BERT, employing the Zero-Shot text classification technique [11]. The "Twitter US Airline Sentiment" dataset is a popular collection of tweets related to US airline companies and the evaluation of their services. This dataset was published on the Kaggle platform and comprises 14,640 tweets, accompanied by comments from each customer who wrote them, the airline company mentioned in each tweet, and the corresponding sentiment category (positive, negative, or neutral). Therefore, each comment is labeled as positive, negative, or neutral. This dataset is frequently utilized in Natural Language Processing and the development of machine learning algorithms for sentiment analysis in text data. We will experimentally explore four different pre-trained Transformers using the Zero-Shot text classification technique to evaluate their performance on an unseen dataset of customer comments for airline companies. The task involves categorizing texts into positive, neutral, and negative sentiment labels, essentially performing Sentiment Analysis. These Transformers have not been previously exposed to or trained specifically on this dataset, making the evaluation more robust and insightful [11]. By investigating how these models respond to the new data, we aim to gain valuable insights into their effectiveness in sentiment analysis tasks and their adaptability to previously unseen contexts.

We will experimentally examine several pre-trained Transformer models to determine if they are effective enough to perform Sentiment classification on three classes using the Zero-Shot technique. For this purpose, we selected a dataset from Kaggle that consists of a total of 14,640 customer comments on airline companies. These comments are divided into 9781 Negative comments, 3099 Neutral comments, and 2363 Positive comments [11]. The following bar plot visually illustrates the distribution of instances based on their category. This dataset does not have a good class distribution or balance; it is imbalanced, which makes the classification task more challenging for any algorithm (Twitter US Airline Sentiment) (**Figure 1**) [12].

So, we are dealing with a quite demanding dataset for any model trained on it. However, we will examine this dataset using the Zero-Shot technique, which means without any training. Therefore, the Transformer models should have a deep understanding of the English language to achieve better results [8].

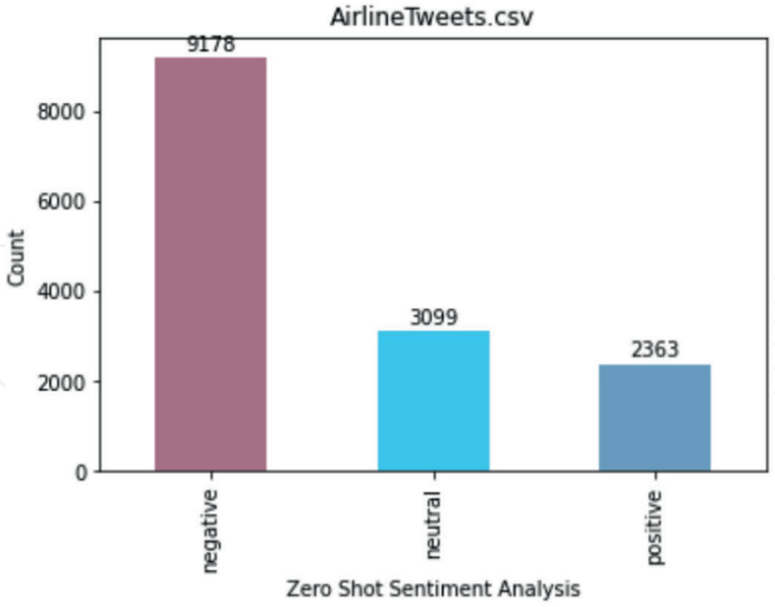

Figure 1.
Bar plot dataset's labels.

The data preprocessing, we performed, was relatively straightforward. We removed all columns that were irrelevant to our purpose, keeping only the column containing the comments and the column with the labels, which represent the actual sentiment ratings (negative, neutral, or positive). We also removed all the names of the airlines. All the other preprocessing steps that we used to do on the texts we wanted to input in the past are now handled by the built-in tokenizer of each Transformer model.

4.2 Tokenizers

4.2.1 BERT tokenizer

The BERT tokenizer is responsible for breaking down the text into smaller units called "tokens." The underlying concept of the BERT tokenizer is to represent the text using a set of tokens that correspond to significant units of the text, such as words or computational symbols.

The tokenizer operates in two main steps. First, it segments the text into words and computational symbols. Then, it converts these words and symbols into unique tokens, each of which is assigned a unique numerical identifier. This transformation allows BERT to operate with inputs of a predetermined size, as each token represents a unit of information [13].

The BERT tokenizer is designed to work in conjunction with the BERT model, creating input that represents the text by utilizing the concept of tokens. Its main function is to represent the text using a set of tokens that correspond to significant units of the text, such as words or computational symbols.

The tokenizer operates in two main steps. First, it segments the text into words and computational symbols. Then, it converts these words and symbols into unique tokens, each of which is assigned a unique numerical identifier. This transformation allows BERT to work with inputs of a fixed size, as each token represents a unit of information.

The BERT tokenizer also includes special functionalities, such as handling special characters (e.g., articles, punctuation marks) and managing the representation of words that exceed the maximum length limit by applying techniques like truncation or padding.

Using the BERT tokenizer, the input text is effectively prepared for processing by the BERT model. It enables the model to understand the meaning of the text and evaluate it in relation to pre-defined categories, without having seen them before [14].

4.2.2 DistilBERT tokenizer

The tokenizer of DistilBERT operates somewhat differently from that of BERT. DistilBERT utilizes a compressed version of BERT with fewer layers and reduced parameters. The tokenizer of DistilBERT follows a similar process as the BERT tokenizer, which involves breaking down the text into smaller units called "tokens." However, due to the reduced number of layers in DistilBERT, its tokenizer performs a simplified tokenization process. This means that the tokens of DistilBERT are fewer in comparison to BERT, and there might be a slight loss of detail in the text representation. Nevertheless, the tokenizer of DistilBERT maintains the fundamental function of the BERT tokenizer, which is to represent the text using tokens [15].

4.2.3 DistilRoBERTa tokenizer

Also, the tokenizer of DistilRoBERTa is different from that of BERT. DistilRoBERTa is based on the RoBERTa model, which is an improved version of BERT. The tokenizer of DistilRoBERTa follows a similar process to the tokenizer of BERT, where the text is broken down into smaller units called "tokens." However, there are some differences in the tokenization rules and token processing. The tokenizer of DistilRoBERTa typically uses a smaller vocabulary compared to BERT, with a limited number of tokens. This results in smaller token representations, but it can still provide high-quality performance in language tasks. Overall, the tokenizer of DistilRoBERTa is adapted to the architecture and requirements of the DistilRoBERTa model for efficiency and effective text processing.

4.3 Transformers

4.3.1 Masked language modeling (MLM)

In order to better understand Transformers and how they work in relation to Sentiment Analysis, we need to grasp one of their fundamental techniques: Masked Language Modeling (MLM).

First and foremost, it is important to know that Transformers have been designed differently depending on the task they aim to accomplish. For instance, when the task at hand is Sentiment Classification or Named Entity Recognition or Question-Answering, suitable Transformers such as BERT, DistilBERT, RoBERTa, and others have been developed specifically for these purposes. On the other hand, when our task involves translation or summarization, appropriate Transformers include Facebook's BART, Google's T5, and others. Similarly, for text generation, models like GPT, GPT2, GPT3, GPT3.5, and GPT-4 utilized by OpenAI, and others are employed.

Masked Language Modeling (MLM) is a technique used in the field of Natural Language Processing (NLP) and Machine Learning to train language models.

In Masked Language Modeling, a randomly selected word or sequence of words in a sentence is hidden (masked), and the model is tasked with predicting what that hidden word or words are. This encourages the model to understand the context and meaning of the surrounding words in order to make the prediction.

For example, a sentence that could be used in an MLM model is as follows: "The big _____ soared through the sky, capturing everyone's attention."

In this case, a word like "bird," "plane," or "kite" could be masked, and the model would need to predict the correct word within the context of the sentence.

Training MLM models is widely known, with BERT (Bidirectional Encoder Representations from Transformers) being one of the most well-known examples. BERT is trained on large bodies of text, where a random portion of words is masked, and the model attempts to predict the correct word based on the context.

Masked Language Modeling models have been successfully used in various applications, such as text completion, information retrieval, and language understanding. The idea is that MLM models can learn from the sequential content of text and reproduce human-like language understanding to a great extent. This ability adds to the model's capacity to classify or characterize texts based on sentiment [16].

4.3.2 Pre-trained models

4.3.2.1 bert-base-uncased

"bert-base-uncased" is a specific pre-trained model variant of BERT (Bidirectional Encoder Representations from Transformers). BERT is a successful machine learning model for Natural Language Processing (NLP) that is trained on large bodies of text to understand the semantic richness of words and sentence structure.

The "bert-base-uncased" version refers to a particular implementation of BERT where words are treated as lowercase (uncased), meaning they are all converted to lowercase. This means that words like "Hello" and "hello" are essentially considered the same by the model.

The difference between "bert-base-uncased" and "BERT" is that "BERT" is a general term referring to the original idea and architecture of the model, while "bert-base-uncased" is a specific implementation of that idea with specific processing parameters.

In general, the designation "bert-base-uncased" is used to describe a specific pre-trained BERT model with certain settings. There are also other variations of BERT, such as "bert-base-cased" (where uppercase and lowercase letters are preserved) and "bert-large-uncased" (a larger model size with more parameters).

As the variations of BERT can have different settings and parameters, it is important to be familiar with the descriptions and documentation to understand precisely what the differences and functionalities of each variation are [13].

4.3.2.2 distilbert-base-cased

"distilbert-base-cased" is a variation of the original BERT (Bidirectional Encoder Representations from Transformers) model that has undergone a process called "distillation" to compress the original model into a smaller size without significant loss in performance.

The differences between "distilbert-base-cased" and the original BERT lie in the following aspects:

1. Model size: "distilbert-base-cased" is significantly smaller than the original BERT. This compression is achieved by reducing the number of model layers, parameters, and nested representation layers.

2. Case sensitivity: Similar to the original BERT, "distilbert-base-cased" maintains the distinction between lowercase and uppercase letters. This means that words like "Hello" and "hello" are considered different by the model.

3. Training with knowledge distillation: The distillation process involves training the "distilbert-base-cased" model using a pre-trained BERT model as a "teacher." The smaller model attempts to replicate the performance of the original model by analyzing the knowledge transferred from the "teacher" model to the "student" model.

The main advantages of "distilbert-base-cased" are its lower memory requirements and computational power compared to the original BERT, making it suitable for applications with limited resources, such as systems with limited memory capacity or low computational power.

Overall, "distilbert-base-cased" is a compressed version of the original BERT that offers reasonably good performance relative to its size compared to the full BERT model, while requiring less space and computational power [15].

4.3.2.3 distilbert-base-uncased-mnli

"distilbert-base-uncased-mnli" is a variation of the BERT (Bidirectional Encoder Representations from Transformers) model that has been trained on the MultiNLI (Multi-Genre Natural Language Inference) dataset.

The differences of "distilbert-base-uncased-mnli" from the original BERT are as follows:

1. Model size: "distilbert-base-uncased-mnli" is compressed and smaller in size compared to the original BERT. This compression is achieved by reducing the number of layers and parameters in the model.

2. Uncased tokens: Similar to the original BERT, "distilbert-base-uncased-mnli" treats all words as uncased, disregarding the distinction between uppercase and lowercase. This means that words like "Hello" and "hello" are considered essentially the same by the model.

3. Training on the MultiNLI dataset: "distilbert-base-uncased-mnli" has been trained on the MultiNLI dataset, which includes pairs of sentences that require evaluating the relationship between them (alternative hypotheses). This trains the model to understand the logical meaning and semantics of the sentences.

Variations of BERT, such as "distilbert-base-uncased-mnli," provide pre-trained models that are adapted to specific domains and datasets. In the case of "distilbert-base-uncased-mnli," it has been specifically trained on the MultiNLI dataset for better performance in logical analysis and evaluating the relationship between sentences.

Overall, "distilbert-base-uncased-mnli" is a compressed variation of the BERT model that has been trained on the MultiNLI dataset. This variation offers a smaller

model size while maintaining the ability to comprehend and evaluate the relationship between sentences [15].

4.3.2.3.1 MultiNLI (multi-genre natural language inference)

The MultiNLI (Multi-Genre Natural Language Inference) dataset is a popular dataset used in the field of Natural Language Processing (NLP) to evaluate the ability of models to understand the meaning and relationship between sentences.

The MultiNLI dataset consists of pairs of sentences known as "hypothesis" and "premise." The "hypothesis" is a statement expressing an idea or hypothesis, while the "premise" is the sentence from which the hypothesis is derived. The main task is to evaluate whether the hypothesis is "entailment," "contradiction," or "neutral" based on the relationship between the two sentences.

MultiNLI encompasses a variety of linguistic materials, covering different genres of literature, scientific texts, news articles, and other types of written material. This ensures the diversity and generalization of the dataset, ensuring that models trained on it can comprehend and respond to various linguistic scenarios.

MultiNLI has been widely used as a dataset for evaluating and training NLP models, including BERT models. Using MultiNLI, we can study a model's ability to understand the meaning of sentences and process the relationships between them.

Overall, MultiNLI represents an important dataset for the development and evaluation of NLP models that deal with recognizing and evaluating the relationship between sentences [17].

4.3.2.4 nli-distilroberta-base

Let us first examine RoBERTa in relation to BERT to understand the version of the pre-trained model nli-distilroberta-base:

RoBERTa is a pre-trained model for Natural Language Processing (NLP) that is a variation of the original BERT (Bidirectional Encoder Representations from Transformers) model. The name "RoBERTa" stands for "Robustly Optimized BERT approach."

The differences between RoBERTa and the original BERT are as follows:

1. Text preprocessing: During text preprocessing, RoBERTa eliminates the case distinction between uppercase and lowercase letters. This means that all letters are converted to lowercase before being processed by the model. This approach allows the model to treat words with different cases as completely different.

2. More training data: RoBERTa is trained on a larger dataset compared to the original BERT. Instead of using 16% of the BERT dataset, RoBERTa utilizes the full datasets of BooksCorpus (800 million words) and CC-News (CommonCrawl News) (76 gigabytes).

3. Training duration: The training algorithm of RoBERTa takes longer than the algorithm used for the original BERT. This means that RoBERTa is trained for more epochs and for a longer period of time to better leverage the available data and improve its performance. These differences constitute enhancements that allow RoBERTa to achieve better results in various NLP tasks compared to the original BERT. However, it is important to note that the fundamental architecture and

ideas that guided BERT remain at the core of RoBERTa, with the differences mainly focusing on training and data preprocessing.

4. Larger model size: RoBERTa has a larger model size compared to the original BERT. This implies that RoBERTa has more parameters and a more detailed representation of words and sentences [18].

The above differences constitute improvements that allow RoBERTa to achieve better results in various NLP tasks compared to the original BERT. However, it is important to note that the basic architecture and ideas that guided BERT remain at the core of RoBERTa, with the differences focusing on training and data preprocessing [18].

Therefore, "nli-distilroberta-base" is a pre-trained model for Natural Language Processing (NLP) that is a variation of the original BERT (Bidirectional Encoder Representations from Transformers) model. This variation utilizes the DistilRoBERTa architecture and has been trained on the NLI (Natural Language Inference) dataset.

The differences of "nli-distilroberta-base" from the original BERT are as follows:

1. Architecture: "nli-distilroberta-base" utilizes the DistilRoBERTa architecture, which is a simplified version of the RoBERTa model. The DistilRoBERTa architecture uses fewer layers and parameters compared to the original BERT, aiming to reduce the size of the model.

2. Training on the NLI dataset: "nli-distilroberta-base" has been trained on the NLI dataset, which consists of sentence pairs for evaluating the relationship between them. Training on the NLI dataset helps the model understand the meaning and relationship between different sentences [11].

5. Experimental results

We will describe the results we obtained from the experimental process of four pre-trained Transformers on the same dataset (Twitter US Airline Sentiment) that we described earlier [12]. The experiments were conducted by us using our own computational resources. The code for the metrics we present was written in Python, utilizing relevant libraries for Transformers with pipelines for the Zero-Shot technique. All Confusion Matrices were generated by combining two functions: confusion_matrix from the sklearn.metrics library and sns.heatmap from the seaborn library.

5.1 Zero-Shot and sentiment analysis distilbert-base-cased

Applying the Zero-Shot technique to the pre-trained Transformer distilbert-base-cased, we obtain the Confusion Matrix (**Figure 2**).

The diagonal of the Confusion Matrix always shows the percentages that the Transformer predicted correctly (**Table 1**). So here we can see that the Transformer correctly predicted 18% of the positive sentiments, 25% of the negative sentiments, and 60% of the neutral sentiments. In the other cells, we can observe the following:

- 23% of the comments that were actually positive were predicted as negative by the model.

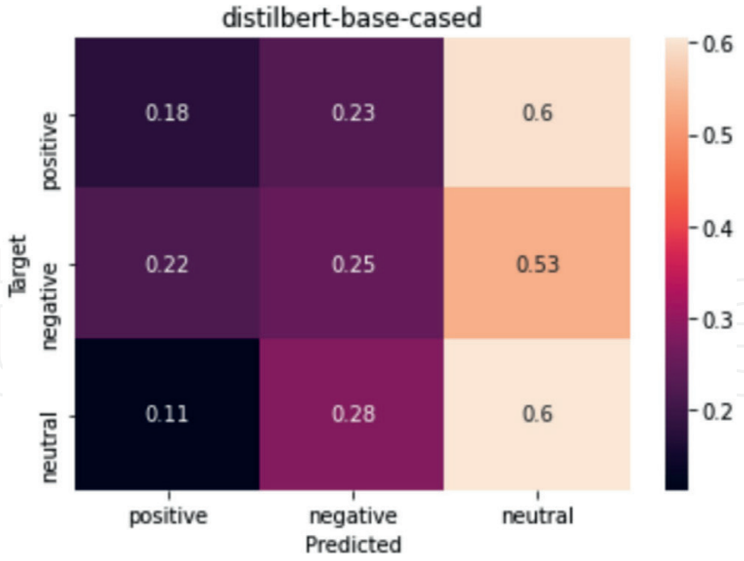

Figure 2.
Confusion Matrix of distilbert-base-cased.

distilbert-base-cased (3 classes)		
Val_accuracy	**f1_score**	**roc_auc_score**
31.14×10^{-2}	31.14×10^{-2}	53.11×10^{-2}

Table 1.
Metrics of distilbert-base-cased.

- 60% of the comments that were actually positive were predicted as neutral by the model.
- 22% of the comments that were actually negative were predicted as positive by the model.
- 53% of the comments that were actually negative were predicted as neutral by the model.
- 11% of the comments that were actually neutral were predicted as positive by the model.
- 28% of the comments that were actually neutral were predicted as negative by the model.
- Val_accuracy = 0.3114071038251366: This metric represents the percentage of correct predictions overall, and we can see that it is approximately 31.14%.
- F1_score = 0.3114071038251366: The F1 score combines precision and recall and measures the balance between them. In your case, the F1 score is also approximately 31.14%.
- Roc_auc_score = 0.531066885655905: The ROC AUC score (Receiver Operating Characteristic Area Under Curve) measures the model's ability to distinguish

between classes. A score of 0.5 represents randomness, while a score of 1 represents perfect discrimination. In your case, the ROC AUC score is approximately 0.531, suggesting a moderate ability to discriminate between classes.

Overall, the model appears to have relatively low performance based on the presented metrics. This could be because the pre-trained model may not have adequately understood such comments, which often contain irony or sarcasm. This does not mean that pre-trained Transformers cannot understand such comments; it means that this specific model has not reached the levels of language comprehension required for use in Zero-Shot Sentiment Classification.

5.2 Zero-Shot and sentiment analysis bert-base-uncased

Applying the Zero-Shot technique to the pre-trained Transformer bert-base-uncased, we obtain the Confusion Matrix (**Figure 3**).

The diagonal of the Confusion Matrix always shows the percentages that the Transformer predicted correctly. So here we can see that the Transformer correctly predicted 58% of the negative sentiments, 17% of the neutral sentiments, and 40% of the positive sentiments. In the other cells, we can observe the following (**Table 2**).

- 16% of the comments that were actually negative were predicted as neutral by the model.
- 26% of the comments that were actually negative were predicted as positive by the model.
- 46% of the comments that were actually neutral were predicted as negative by the model.
- 36% of the comments that were actually neutral were predicted as positive by the model.

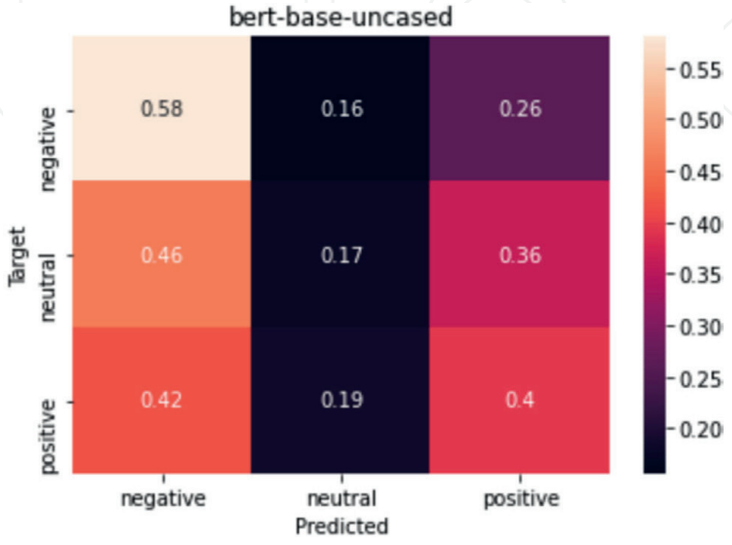

Figure 3.
Confusion Matrix of bert-base-uncased.

bert-base-uncased (3 classes)		
Val_accuracy	**f1_score**	**roc_auc_score**
46.45×10^{-2}	46.45×10^{-2}	50.83×10^{-2}

Table 2.
Metrics of bert-base-uncased.

- 42% of the comments that were actually positive were predicted as negative by the model.
- 19% of the comments that were actually positive were predicted as neutral by the model.

Based on the metrics provided for the Transformer bert-base-uncased for sentiment analysis with Zero-Shot text classification, we can draw the following conclusions:

The validation accuracy is low, at 0.4644808743169399. This indicates that the model struggles in recognizing the three classes in the dataset.

The F1 score for the model is 0.4644808743169399, representing the harmonic mean of precision and recall. This suggests that the model has limited performance in both precision and recall.

The ROC AUC score is 0.5083023131851064, which is low. This indicates that the model has limited ability to correctly distinguish the three classes.

5.3 Zero-Shot and sentiment analysis distilbert-base-uncased-mnli

Applying the Zero-Shot technique to the pre-trained Transformer distilbert-base-uncased-mnli, we obtain the Confusion Matrix (**Figure 4**).

The diagonal of the Confusion Matrix always shows the percentages that the Transformer predicted correctly. So here we can see that the Transformer correctly

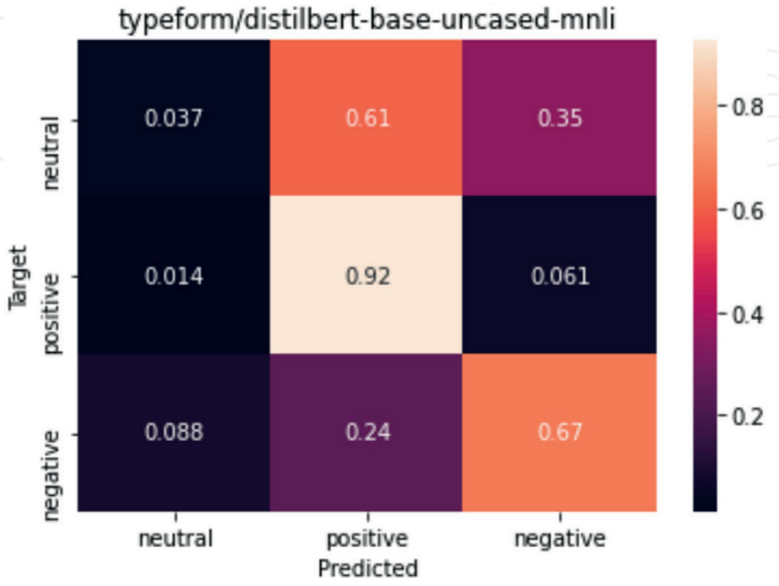

Figure 4.
Confusion Matrix of distilbert-base-uncased-mnli.

typeform/distilbert-base-uncased-mnli (3 classes)		
Val_accuracy	**f1_score**	**roc_auc_score**
57.61×10^{-2}	57.61×10^{-2}	76.83×10^{-2}

Table 3.
Metrics of distilbert-base-uncased-mnli.

predicted 3.7% of the neutral sentiments, 92% of the positive sentiments, and 67% of the negative sentiments. In the other cells, we can observe the following (**Table 3**).

- 61% of the comments that were actually neutral were predicted as positive by the model.
- 35% of the comments that were actually neutral were predicted as negative by the model.
- 1.4% of the comments that were actually positive were predicted as neutral by the model.
- 6.1% of the comments that were actually positive were predicted as negative by the model.
- 8.8% of the comments that were actually negative were predicted as neutral by the model.
- 24% of the comments that were actually negative were predicted as positive by the model.

Based on the metrics provided for the Transformer distilbert-base-uncased-mnli for sentiment analysis with Zero-Shot text classification, we can draw the following conclusions:

Validation accuracy (Val_accuracy): The validation accuracy is 0.576. This means that the Transformer correctly classifies the sentiment of the text into three categories (3 classes) with an average accuracy of 57.6%. This accuracy indicates that the Transformer has a relatively moderate performance, and there is room for improvement.

F1 score: The F1 score is 0.576, which is equal to the validation accuracy. The F1 score is a measure of the overall performance that combines precision and recall. The value of 0.576 indicates that the Transformer has a moderate performance and needs improvement in this area.

ROC AUC score: The ROC AUC score is 0.768. This metric evaluates the model's ability to distinguish between classes and correctly rank examples based on the predicted probabilities. A ROC AUC score of 0.768 indicates that the Transformer has a relatively good discriminative ability between classes, but there is still room for improvement.

Overall, we can say that the Transformer distilbert-base-uncased-mnli has a moderate performance in sentiment analysis with Zero-Shot text classification, and there is room for improvement in terms of accuracy and F1 score. However, the ability to distinguish between classes, as represented by the ROC AUC score, is relatively good.

5.4 Zero-Shot and sentiment analysis nli-distilroberta-base

Applying the Zero-Shot technique to the pre-trained Transformer nli-distilroberta-base, we obtain the Confusion Matrix (**Figure 5**).

The diagonal of the Confusion Matrix always shows the percentages that the Transformer predicted correctly. So here we can see that the Transformer correctly predicted 4.4% of the neutral sentiments, 87% of the positive sentiments, and 86% of the negative sentiments. In the other cells, we can observe the following (**Table 4**).

- 39% of the comments that were actually neutral were predicted as positive by the model.
- 57% of the comments that were actually neutral were predicted as negative by the model.
- 2.0% of the comments that were actually positive were predicted as neutral by the model.
- 11% of the comments that were actually positive were predicted as negative by the model.
- 3.3% of the comments that were actually negative were predicted as neutral by the model.

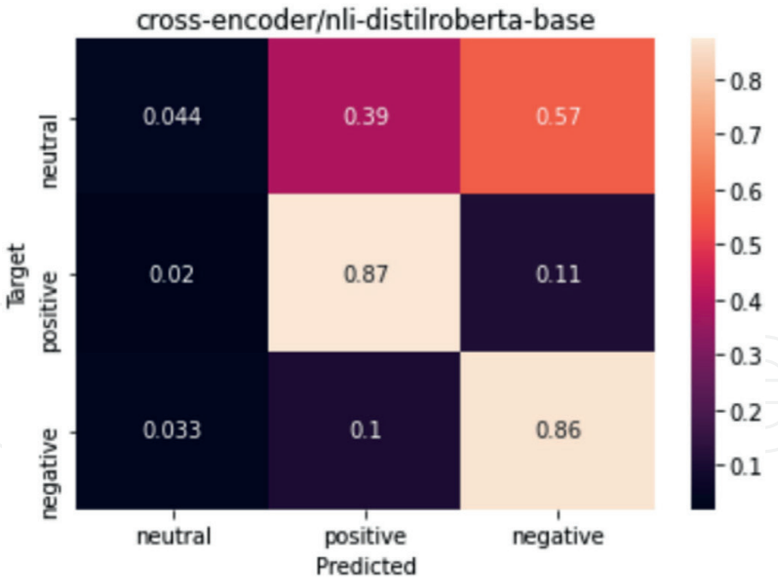

Figure 5.
Confusion Matrix of nli-distilroberta-base.

cross-encoder/nli-distilroberta-base (3 classes)		
Val_accuracy	**f1_score**	**roc_auc_score**
69.17×10^{-2}	69.17×10^{-2}	82.63×10^{-2}

Table 4.
Metrics of nli-distilroberta-base.

- 10% of the comments that were actually negative were predicted as positive by the model.

Based on the metrics provided for the Transformer nli-distilroberta-base, which performs sentiment analysis using Zero-Shot text classification, we can draw the following conclusions:

1. Validation accuracy (Val_accuracy): The validation accuracy is 0.692. This indicates that the Transformer correctly classifies the sentiment of the text into three categories (3 classes) with an accuracy of 69.2%. This accuracy shows a relatively good performance, suggesting that the Transformer is effective in predicting sentiment.

2. F1 score: The F1 score is 0.692, which is equal to the validation accuracy. The F1 score combines precision and recall and provides an overall measure of performance. The value of 0.692 indicates that the Transformer has a relatively good balance between precision and recall, resulting in accurate predictions for sentiment analysis.

3. ROC AUC score: The ROC AUC score is 0.826. This metric evaluates the model's ability to differentiate between classes and rank examples based on predicted probabilities. A ROC AUC score of 0.826 indicates that the Transformer has a good discriminative ability, with a high likelihood of correctly distinguishing between different sentiment classes.

In summary, the Transformer nli-distilroberta-base demonstrates a relatively good performance in sentiment analysis using Zero-Shot text classification. It achieves high accuracy, F1 score, and ROC AUC score, indicating its effectiveness in accurately predicting sentiment and distinguishing between different sentiment classes (**Table 5** and **Figure 6**).

Based on the overall table for the performances of the Transformers we have and the bar plot, we can draw the following conclusions:

Validation accuracy: The models significantly differ in validation accuracy. The nli-distilroberta-base model has the highest validation accuracy at around 69.2%, while the distilbert-base-cased model has the lowest validation accuracy at around 31.1%.

F1 score: Similar to the validation accuracy, the nli-distilroberta-base model achieves the highest F1 score at around 69.2%, while the distilbert-base-cased model has the lowest F1 score at around 31.1%.

ROC AUC score: The nli-distilroberta-base model has the highest ROC AUC score at around 82.6%, indicating a good discriminative ability between classes.

Transformers	Validation accuracy	F1 score	ROC AUC score
distilbert-base-cased	31.14×10^{-2}	31.14×10^{-2}	53.11×10^{-2}
bert-base-uncased	46.45×10^{-2}	46.45×10^{-2}	50.83×10^{-2}
distilbert-base-uncased-mnli	57.61×10^{-2}	57.61×10^{-2}	76.83×10^{-2}
nli-distilroberta-base	69.17×10^{-2}	69.17×10^{-2}	82.63×10^{-2}

Table 5.
Comparison in the metrics of all models.

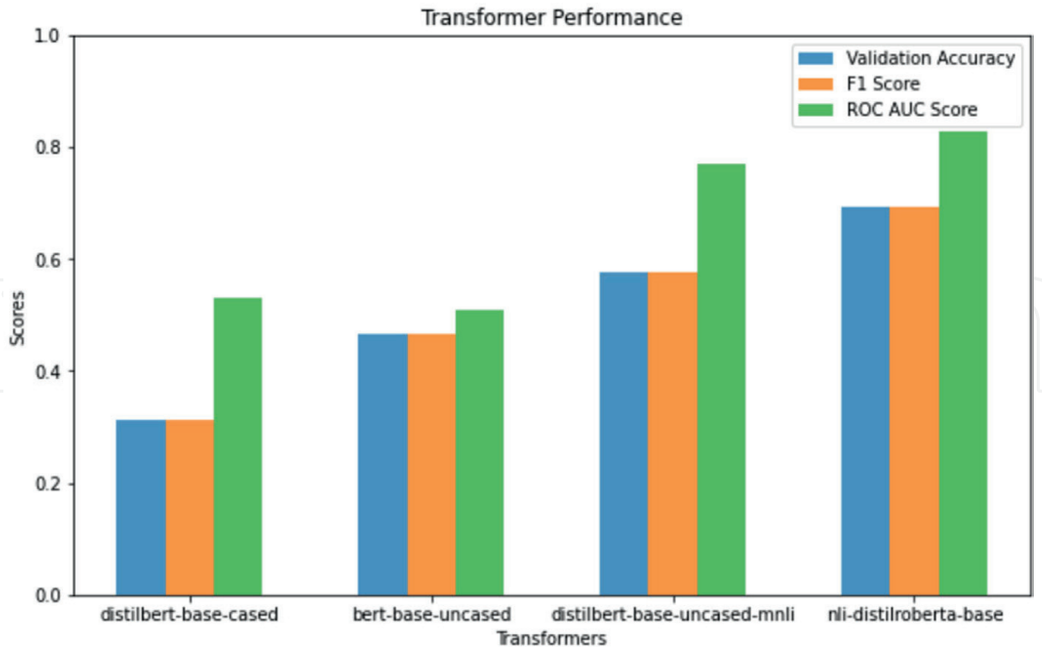

Figure 6.
Bar plot comparison of all models.

On the contrary, the distilbert-base-cased model has the lowest ROC AUC score at around 53.1%.

Overall, the nli-distilroberta-base model stands out among the other three models in all metrics. It demonstrates higher validation accuracy, F1 score, and ROC AUC score compared to the other models. On the other hand, the distilbert-base-cased model shows the lowest performance across all metrics.

Therefore, we can conclude that the nli-distilroberta-base model is the most effective among the four models examined for sentiment analysis.

6. Conclusions

The fact that we followed the Zero-Shot Sentiment Classification technique limits us in terms of fine-tuning to achieve optimal results in Sentiment Analysis for this specific dataset. Through these experiments, a new technique is highlighted, which can be applied to vast datasets. With the Zero-Shot technique, we can achieve Sentiment Classification without human supervision. One might wonder how many human hours are required to evaluate a massive dataset without errors. This method can be likened to unsupervised learning. Furthermore, it is another approach to understand how well the pre-trained Model has immersed itself in the language.

The percentages we achieved in the experiments demonstrate to what extent this particular Transformer Model has been trained on similar data and how well it has understood the language. Such an effort to apply the Zero-Shot Sentiment Classification technique on the Twitter US Airline Sentiment dataset has not been done before, so there is no comparative reference we can provide. However, works have been done with this technique in other domains, such as a study proposing a method for conducting Zero-Shot Aspect-Based Sentiment Analysis without using domain-specific training data, among others [19].

The abundance of user-generated information on the Web necessitates accurate methods for analyzing and determining users' opinions and attitudes toward events, products, and entities. In this study, we designed and implemented BERT-like Transformers for the task of Zero-Shot classification. These four pre-trained Transformer models deliver commendable results, despite having only a few million parameters.

Future work will focus on several directions based on the presented results in this chapter using the Zero-Shot technique. First, exploring other models known for their performance in sentiment analysis with the Zero-Shot technique should be considered. Evaluating their accuracy, F1 score, and ROC AUC score and comparing them to those of the existing models will be beneficial. Experimenting with different model variations to identify the most suitable one for specific requirements is recommended. Additionally, examining data preprocessing techniques and evaluating the steps involved in data preprocessing should be conducted. Lastly, exploring ensemble models that combine multiple models to enhance performance can be advantageous. The utilization of diverse models can offer improvements in terms of accuracy and overall performance. These are potential avenues for future work to enhance the results of sentiment analysis using the Zero-Shot techniques.

Acknowledgements

We would like to express our gratitude to Mr. Panagiotis Hadjidoukas from the Department of Computer Logic, part of the Department of Computer Engineering & Informatics at the University of Patras, for providing us with the computational resources necessary to conduct these demanding experiments.

This work was partially supported by the Project entitled "Strengthening the Research Activities of the Directorate of Infrastructure and Networks," funded by the Computer Technology Institute and Press "Diophantus" with project code 0822/001.

Author details

Konstantinos Kyritsis[1,2], Nikolaos Spatiotis[1,2], Isidoros Perikos[1,2,3] and Michael Paraskevas[1,2]*

1 Computer Technology Institute and Press "Diophantus," Patras, Greece

2 Electrical and Computer Engineering Department, University of Peloponnese, Greece

3 Computer Engineering and Informatics Department, University of Patras, Greece

*Address all correspondence to: mparask@cti.gr

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. New York: Cornell University Library in Ithaca; 2017. Available from: https://arxiv.org/abs/1706.03762

[2] Prottasha NJ, Sami AA, Kowsher M, Murad SA, Bairagi AK, Masud M, et al. Transfer learning for sentiment analysis using BERT based supervised fine-tuning. Sensors. 2022;**22**(11):4157

[3] Chi S, Luyao H, Xipeng Q. Utilizing BERT for aspect-based sentiment analysis. arxiv.org. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1; 2019. pp. 380-385. Available from: https://aclanthology.org/N19-1035/

[4] Zhang T, Gong X, Chen CLP. BMT-net: Broad multitask transformer network for sentiment analysis. IEEE Access. 2022;**52**(7):6232-6243. Available from: https://ieeexplore.ieee.org/document/9369997

[5] Cheng J, Fostiropoulos I, Boehm B, Soleymani M. Multimodal phased transformer for sentiment analysis. In: Conference on Empirical Methods in Natural Language Processing. United States: Association for Computational Linguistics (ACL); 2021. Available from: https://aclanthology.org/2021.emnlp-main.189/

[6] Tan KL, Lee CP, Lim KM, Anbananthen KSM. Sentiment analysis with ensemble hybrid deep. IEEE Access. 2022;**10**:103694-103704. Available from: https://doaj.org/article/948b7ca90291416fb31bda6b789b8920

[7] Tesfagergish SG, Kapočiūtė-Dzikienė J, Damaševičius R. Zero-Shot emotion detection for semi-supervised sentiment analysis using sentence transformers and ensemble learning. Applied Sciences. 2022;**12**(17):8662

[8] Yang P, Wang J, Gan R, Zhu X, Zhang L, Wu Z, et al. Zero-Shot learners for natural language understanding via a unified multiple choice perspective. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Pennsylvania, United States: Association for Computational Linguistics (ACL); 2022

[9] Yin W, Hay J and Roth D. Benchmarking Zero-Shot text classification: Datasets, evaluation and entailment approach. In: IJCNLP 2019. Pennsylvania, United States: Association for Computational Linguistics (ACL); 2019. Available from: https://aclanthology.org/D19-1404/

[10] Pushp PK, Srivastava MM. Train once, test anywhere: Zero-Shot learning for text classification. arXiv: Computation and Language. 2017. [preprint]

[11] Delangue C, Chaumond J, Wolf T. Hugging Face [Online]. U.S.: Hugging Face, Inc.; 2016. Available from: https://huggingface.co/

[12] Sculley D, Elliott J, Hamner B, Moser J. Kaggle [Online]. 2010. Available from: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[13] Devlin J, Ming-Wei C, Kenton L and Kristina T. BERT: Pre-training of deep bidirectional transformers for language understanding. United States: Association for Computational

Linguistics (ACL); 2019. DOI: 10.48550/arXiv.1810.04805.

[14] Kudo T, Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Empirical Methods in Natural Language Processing: System Demonstrations. Pennsylvania, United States: Association for Computational Linguistics (ACL); 2018

[15] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In: 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS 2019. Vancouver; 2020

[16] Salazar J, Liang D, Nguyen TQ, Kirchhoff K. Masked language model scoring. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics [Online]. Pennsylvania, United States: Association for Computational Linguistics (ACL); 2020. Available from: https://aclanthology.org/2020.acl-main.240/

[17] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research. 2020;**21**:1-67

[18] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A robustly optimized BERT Pretraining approach. In: Proceedings of the 20th Chinese National Conference on Computational Linguistics. Huhhot; 2021

[19] Shu L, Xu H, Liu B, Chen J. Zero-Shot Aspect-Based Sentiment Analysis. 2022. Available from: https://arxiv.org/pdf/2202.01924.pdf