



Article

Public Perception of ChatGPT and Transfer Learning for Tweets Sentiment Analysis Using Wolfram Mathematica

Yankang Su ^{1,*}  and Zbigniew J. Kabala ² ¹ Pioneer Academics, 101 Greenwood Ave, Ste 170, Jenkintown, PA 19046, USA² Department of Civil & Environmental Engineering, Duke University, Durham, NC 27708, USA; zbigniew.kabala@duke.edu

* Correspondence: suxin@sxu.edu.cn

Abstract: Understanding public opinion on ChatGPT is crucial for recognizing its strengths and areas of concern. By utilizing natural language processing (NLP), this study delves into tweets regarding ChatGPT to determine temporal patterns, content features, and topic modeling and perform a sentiment analysis. Analyzing a dataset of 500,000 tweets, our research shifts from conventional data science tools like Python and R to exploit Wolfram Mathematica's robust capabilities. Additionally, with the aim of solving the problem of ignoring semantic information in the LDA model feature extraction, a synergistic methodology entwining LDA, GloVe embeddings, and K-Nearest Neighbors (KNN) clustering is proposed to categorize topics within ChatGPT-related tweets. This comprehensive strategy ensures semantic, syntactic, and topical congruence within classified groups by utilizing the strengths of probabilistic modeling, semantic embeddings, and similarity-based clustering. While built-in sentiment classifiers often fall short in accuracy, we introduce four transfer learning techniques from the Wolfram Neural Net Repository to address this gap. Two of these techniques involve transferring static word embeddings, "GloVe" and "ConceptNet", which are further processed using an LSTM layer. The remaining techniques center on fine-tuning pre-trained models using scantily annotated data; one refines embeddings from language models (ELMo), while the other fine-tunes bidirectional encoder representations from transformers (BERT). Our experiments on the dataset underscore the effectiveness of the four methods for the sentiment analysis of tweets. This investigation augments our comprehension of user sentiment towards ChatGPT and emphasizes the continued significance of exploration in this domain. Furthermore, this work serves as a pivotal reference for scholars who are accustomed to using Wolfram Mathematica in other research domains, aiding their efforts in text analytics on social media platforms.

Keywords: ChatGPT; Twitter; topic modeling; sentiment analysis; transfer learning

Citation: Su, Y.; Kabala, Z.J. Public Perception of ChatGPT and Transfer Learning for Tweets Sentiment Analysis Using Wolfram Mathematica. *Data* **2023**, *8*, 180. <https://doi.org/10.3390/data8120180>

Academic Editor: Han Woo Park

Received: 7 September 2023

Revised: 24 October 2023

Accepted: 21 November 2023

Published: 28 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The widespread use of social media provides a wealth of data for public sentiment analyses. While artificial intelligence (AI), especially ChatGPT, is rapidly advancing and becoming integrated into society [1,2], our understanding of people's attitudes towards these innovative technologies has not kept pace. Previous research and the literature have emphasized the importance of assessing user sentiment towards newly introduced AI services. Using social media data, researchers can explore public opinions on AI through NLP-based approaches. Understanding user sentiments is crucial as they can provide insights into the potential success or failure of a technology as well as its strengths and weaknesses. Assessing the overall sentiment towards ChatGPT can provide valuable insights into the product's potential success or failure. Exploring user attitudes from popular social media platforms may reveal whether these assessments align with actual user experiences. In conclusion, further examination of ChatGPT's user sentiment is essential. To investigate the sentiments of ChatGPT users, we analyzed data from Twitter, a

platform that enables users to read and share short messages called “tweets”. With Twitter’s growing popularity, researchers and practitioners are increasingly turning to Twitter data to glean valuable insights from potential customers [3–5].

In our study, we leverage natural language processing (NLP) techniques such as the analysis of temporal tweet volume trends, content feature analyses, topic modeling, and sentiment analyses to understand the public’s view of ChatGPT using a dataset comprising 500,000 tweets. For this analysis, we adopted the Wolfram Mathematica software, a contemporary computational science platform that spans a wide range of areas including algebra, applied mathematics, computer science, earth sciences, engineering, data science (DS), social sciences, and more. While Python and R remain the predominant tools in the realm of DS research, Wolfram Mathematica, with its abundant functions and integrated features, stands as a valuable platform and approach. Within Mathematica, all functionalities are available from the outset, eliminating the need for additional packages. Although most tasks require no extra packages, an unofficial package repository is available for those who seek it. Considering the limited research publications on tracking public attitudes on Twitter via Wolfram Mathematica and on employing transfer learning based on the Wolfram Neural Net Repository for tweet sentiment analysis, this paper introduces a significant novelty in the domain. Furthermore, this work serves as a salient reference for scholars proficient in Wolfram Mathematica, facilitating their endeavors in text analytics on social media platforms. Except for a few images for which sources are provided, all other illustrations in this paper were computed and generated using the Wolfram Mathematica Notebook.

Additionally, aiming at solving the problem of ignoring semantic information in the LDA model feature extraction, a synergistic methodology entwining LDA, GloVe embeddings, and K-Nearest Neighbors (KNN) clustering is proposed to categorize topics within ChatGPT-related tweets. This comprehensive strategy ensures semantic, syntactic, and topical congruence within classified groups by utilizing the strengths of probabilistic modeling, semantic embeddings, and similarity-based clustering.

To tackle the challenge of low accuracy observed with built-in classifiers in tweet sentiment analyses, we have introduced four transfer learning methods employing deep learning, which are grounded in the Wolfram Neural Net Repository. More specifically, two of the methods involved separately transferring diverse static word embeddings (“GloVe” and “ConceptNet”) and then processing them in a long short-term memory (LSTM) layer. The others involved fine-tuning pre-trained models with limited annotated data, known as fine-tuning embeddings from language models (ELMo) and fine-tuning bidirectional encoder representations from transformers (BERT). Our experiments on the dataset underscore the effectiveness of the four techniques for the sentiment analysis of ChatGPT-related tweets. This research advances our comprehension of user sentiment towards ChatGPT and emphasizes the need for continued investigation into this subject matter.

2. Related Works

This study was informed by research articles from multiple disciplines. Therefore, in this section, we cover the literature review of textual analytics of Twitter, NLP, sentiment analysis, and machine learning, particularly deep learning methods.

2.1. Tweets Analytics and Topic Modeling

Recently, there has been an increase in interest in sentiment analysis techniques, driven by the increasing need for tools to monitor public sentiment among end users [6]. This trend is further evidenced by the rising number of companies offering brand tracking and marketing services, which now often incorporate sentiment analysis as part of their packages [7]. A recent study exploring sentiments of ChatGPT users relied solely on 10,732 tweets from early users collected between 5 and 7 December 2022 [8]. It is important to note that sentiment analysis users should be wary of drawing conclusions based on data from narrow sources like Twitter and should instead focus on collecting data from a broad spectrum of users with diverse demographic and personality traits.

With the launch of ChatGPT, there has been a surge in research interest in understanding the public's perception of this technology. Numerous studies have explored the wide-reaching societal impacts [9] and domain-specific applications [10] of ChatGPT. Most of these studies relied on social science methods such as interviews, user experience evaluations, and expert insights. However, a few studies, more aligned with our approach, used computational techniques to explore public sentiment towards key themes related to ChatGPT using social media data.

Haque et al. utilized latent Dirichlet allocation (LDA) topic modeling to uncover prominent topics in tweets related to ChatGPT [8]. They then performed a sentiment analysis on these topics through manual labeling. This per-topic sentiment analysis revealed the varied perceptions of early ChatGPT users towards different topics. Nevertheless, due to the restricted dataset and limited duration of their study, the study was unable to provide a comprehensive view of the public sentiment towards ChatGPT. Similarly, Taecharungroj employed LDA topic modeling on a larger dataset of over 200,000 tweets, but the study's primary focus was on ChatGPT's capabilities and shortcomings, and it did not delve into a sentiment analysis or public opinion exploration [11].

2.2. Deep Learning for Tweets Sentiment Analysis

In recent years, sentiment analysis on social media platforms, notably Twitter, has garnered increasing research attention. Analyzing sentiments on these platforms has emerged as a pivotal method for understanding public attitudes towards specific domains or topics [12–15]. Existing research on Twitter sentiment analysis is primarily divided into two categories: supervised techniques [16–18] and lexicon-driven approaches [19,20]. The supervised approaches utilize classifiers such as naive Bayes, support vector machine, and random forest. These techniques harness different feature combinations, including part-of-speech (POS) tags, word N-grams, and contextual tweet details like hashtags, retweets, emoticons, and capitalized words. In contrast, lexicon-driven approaches leverage precompiled word lexicons, which are associated with sentiment scores. These strategies typically depend on overt lexical or syntactical markers that convey sentiment. However, the sentiment of many tweets is often embedded in the contextual semantics.

With the advancements that have been made in machine learning, deep learning techniques have gained prominence, especially in tasks involving image and voice processing. In recent times, these techniques have started to surpass conventional linear models in natural language processing (NLP). A notable example is the work by Kalchbrenner et al. [21], where the authors introduced a multi-layered convolutional neural network. This network uses latent, densely packed, low-dimensional word vectors as inputs. They employed this dynamic convolutional network to classify sentiments in movie reviews and tweets, proving its superiority over unigram and bigram models. Furthering this approach, dos Santos et al. [22] designed a deep convolutional neural network that captures information ranging from the character to sentence level with the aim of sentiment classification for concise texts. Although these models are potent, they require intricate feature crafting. Additionally, while feature engineering results in sparse vectors, deep learning methods yield richer dense vectors. However, crafting these dense vectors necessitates vast training datasets.

Leveraging pre-trained dense vectors imbued with both syntactic and semantic knowledge can drastically enhance performance in deep learning-driven NLP tasks. For instance, Kim [23] employed embeddings initialized with word vectors pre-trained on a massive dataset from Google News. Similarly, Zhang et al. [24] used a variety of pre-trained word embeddings (such as Word2Vec, GloVe, and syntactic embedding) and applied convolutional neural networks (CNNs) to each set independently. The resulting feature vectors from each embedding set were then fused to form a comprehensive feature vector. The idea of transferring pre-acquired semantic and syntactic knowledge from varied tasks has gained traction in the NLP community. Given their foundational role, pre-trained word embeddings can significantly elevate the efficacy of deep learning endeavors in NLP.

Utilizing generalized embeddings—be it at the word, sentence, or paragraph level—has also found applications in downstream tasks like sentiment analysis, text categorization, text clustering, and translation.

3. Public Perception of ChatGPT on Twitter

3.1. Research Objectives and Questions

Research Objectives

The overarching goal of this study was to delve into the public perception of ChatGPT on Twitter. We aimed to understand:

- The dynamics and trends related to discussions involving ChatGPT;
- The nature and characteristics of the content in these discussions;
- The primary topics and sentiments prevalent in conversations about ChatGPT on Twitter.

Research Questions

- RQ1: How have the trends in the number of tweets mentioning ChatGPT evolved over the studied time span?

This question will allow us to elucidate whether there have been fluctuations in public interest and engagement with ChatGPT on Twitter.

- RQ2: What are the overall characteristics of the content of tweets mentioning ChatGPT?

This seeks to explore the sentiments, tone, and lexicon prevalent in tweets, providing insights into the public's attitudes and opinions on ChatGPT.

- RQ3: What are the main topics that are being discussed about ChatGPT on Twitter?

The aim is to identify the prevalent themes and subjects in the discussions, uncovering which aspects of ChatGPT Twitter users are most focused on.

3.2. Method

3.2.1. Data Source

Because our research primarily focuses on “Public Perception of ChatGPT on Tweets,” it necessitates the use of a ChatGPT-centric tweets dataset. Our thorough search on Kaggle and other open-source dataset platforms revealed a limited number of specialized datasets pertaining to this topic. We assessed these available datasets in terms of scale, temporal span, statistical properties, and user reviews, ultimately selecting the most suitable one for our research. The chosen dataset spans several months immediately following the release of ChatGPT, a period notable for its dense clustering of ChatGPT-related news events. The tweets therein reflect the fluctuations and variations in early user attitudes, rendering the dataset highly representative and thus of significant research value.

We openly acknowledge the partial limitations of our research dataset which, while revealing and significant within the context of the datasets analyzed, might be subject to certain limitations regarding generalizability due to the specificity and temporal nature of the datasets utilized. Moreover, we will suggest avenues for future research that might explore similar objectives using alternative datasets and timeframes. In light of the above, while we recognize the importance of utilizing expansive and varied datasets, the specific focus and contextual nuances of our study guided our dataset selection to lean towards what is most applicable and available.

The target dataset for the study presented in this paper is “500 k ChatGPT-related Tweets Jan-Mar 2023,” which was sourced from the Kaggle website. This dataset contains a CSV file related to ChatGPT including keywords (“chatgpt”, “gpt”), “#hashtags”, and “@mentions” about ChatGPT. The file includes information on 500,000 tweets. The dataset aims to help understand public opinion, trends, and potential applications of ChatGPT by analyzing tweet volume, sentiment, user engagement, and the influence of key AI events. The dataset offers valuable insights for companies, researchers, and policymakers, allowing them to make informed decisions and shape the future of AI-powered conversational technologies. The collected data spans from 4 January 2023 to 29 March 2023 (almost three

months), providing ample time to observe daily and weekly trends. The work presented in this section is inspired by the insights from the research conducted by scholar Khalid Ansari [25]. The dataset has the following columns:

- date: date of the tweet post;
- id: a unique identifier of the tweet;
- content: actual tweet content;
- username: username of the Twitter user;
- like_count and retweet count: like and retweet counts for that tweet.

Data procurement was executed utilizing a Python module, *snsrape*, which was designed to facilitate the extraction of extensive data from social networking sites with minimal lines of code, overcoming the limitations and restrictions associated with conventional data collection tools like Twitter API. This module surpasses the constraints imposed by Twitter API, such as rate limits and access to historical data, enabling the retrieval of tweets older than the standard 7-day limit allowed by the free tier of Twitter API. Nonetheless, it is crucial to utilize *snsrape* judiciously to prevent overloading Twitter servers and to comply with the platform's terms of service.

In [25], methodologies are meticulously outlined for the extraction of Twitter data leveraging parameters like keywords, languages, date ranges, and usernames. Such parameters can be integrated in diverse combinations as a query string into the scraper. A query may consist of a singular keyword or a composite of keywords or can employ advanced search expressions to streamline results according to predefined conditions such as date ranges, hashtags, mentions, etc. The "OR" condition was applied to amalgamate multiple query strings, keywords (chatgpt, chat gpt), hashtags (#chatgpt), and mentions (@chatgpt), ensuring the capture of a substantial quantity of pertinent ChatGPT tweets.

3.2.2. Data Pre-Processing

In this study, several data cleaning techniques were utilized, as follows:

Negation handling: due to the significance of negations in discerning tweet sentiment, common abbreviations like "won't", "can't", and "n't" were transformed into their full forms: "will not", "cannot", and "not".

URL elimination: given the general consensus that URLs minimally impact sentiment, shortened Twitter URLs were expanded, tokenized, and subsequently removed from the tweets.

Elongated word correction: to handle the common trend of word elongation in tweets, sequences of characters repeated more than thrice consecutively were truncated to three repetitions, differentiating standard words from their elongated variants.

Numerical value removal: numerical values, often deemed irrelevant for sentiment analysis, were omitted to enhance content clarity.

Stop word filtration: recognizing the potential interference of frequently occurring words in sentiment analysis, such words, or "stop words", were filtered out based on predefined lists.

3.2.3. Tweets Exploration

Visualizing tweet volume against the dates will give us richer insights into user engagement (tweet volume) on ChatGPT based on major events surrounding the technology. To address the RQ1, we draw a histogram to glean the daily and weekly trends based on the volume of tweets (Figure 1). In the subsequent "Results and Analysis" section of this chapter, we will delve into the temporal trends of the tweets and analyze the events associated with them.

To address RQ2, we can use word clouds. By utilizing word clouds, we can effectively provide a snapshot of the primary topics and connections in the ongoing ChatGPT discussions on Twitter. **Uni-grams (words):** This gives us a visual of the most common topics and applications surrounding ChatGPT. For uni-grams, we first have to lemmatize, which is a process of reducing words to their base form called a lemma, allowing us to group similar

word forms together. We use the Twitter old logo for the word cloud mask to make the results look relevant to the project (Figure 2).

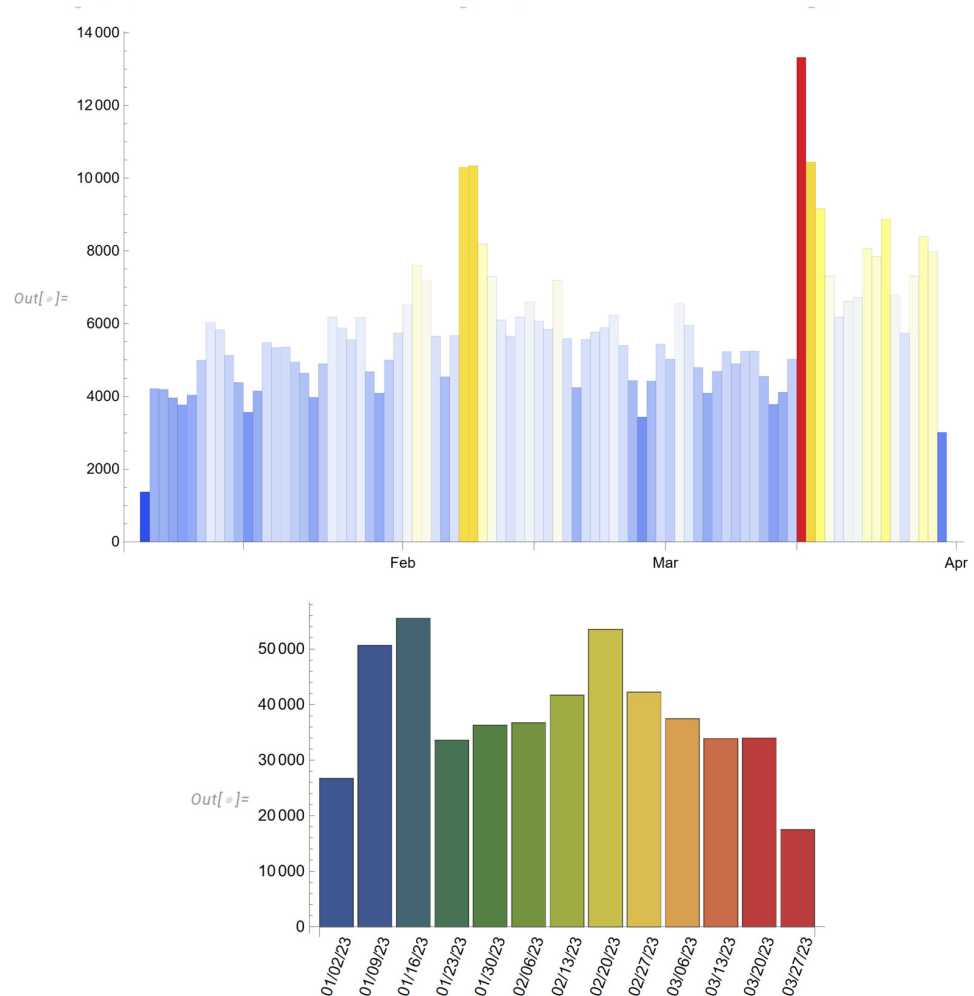


Figure 1. Trends in daily and weekly tweet counts within the dataset.

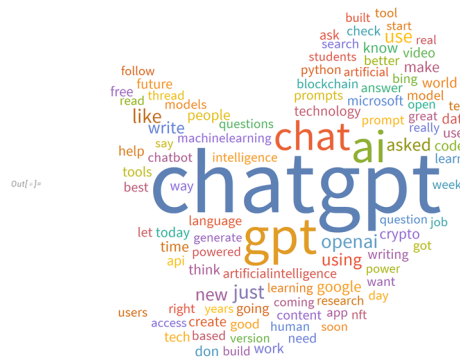


Figure 2. Word cloud of tweets.

3.2.4. Topic Classification and Modeling

To address RQ3, we employ both Wolfram’s built-in classifier and the LDA approach for the topic classification and modeling of the top 10,000 tweets with the highest number of likes related to ChatGPT.

Wolfram Mathematica Built-in Classifier

The Wolfram Mathematica built-in classifier has the following features:

- This classifier attempts to infer the topic of a text;
- This classifier has been trained on Facebook posts but can be used for other texts as well;
- The input text should typically be one or a few sentences;
- The current version only works for the English language.

We utilized the built-in classifier to categorize the topics of the tweets. Subsequently, we visualized the classification results using a bar chart, where the x-axis denotes the names of each category and the y-axis represents the number of tweets pertaining to the respective topics (Figure 3).

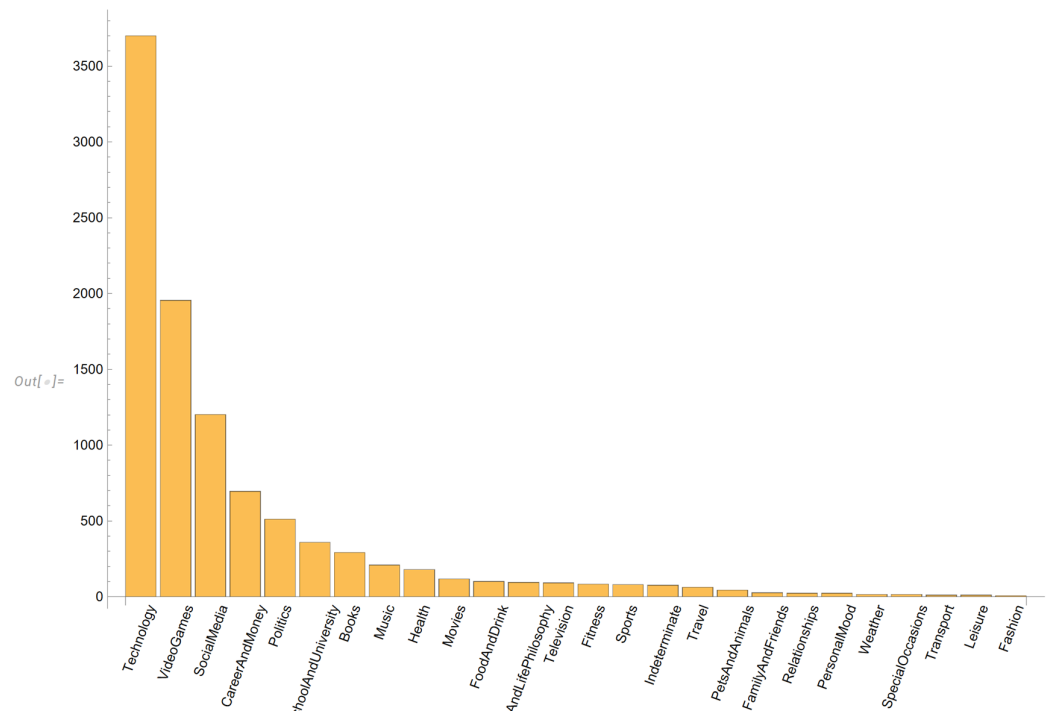


Figure 3. Topic classification of tweets with the built-in classifier of Wolfram Mathematica.

LDA Topic Modeling

In our topic modeling endeavor, we employed the LDA methodology. LDA, an unsupervised technique, categorizes documents analogously to numeric clustering, which is useful for discerning intrinsic groupings without predefined criteria. It conceptualizes each document as a mix of topics and each topic as an array of words, allowing for shared content across tweets rather than rigid categorizations.

Though Wolfram language lacks an innate LDA function, Wolfram Mathematica Notebook facilitates Python code execution. We can use an external evaluation with Python and its libraries to extend the Wolfram Language instantly. Therefore, we integrated Python within our Wolfram framework, leveraging the “gensim” Python library for the LDA topic modeling of tweets. When applying LDA to the top 10,000 favored tweets, text refinement is paramount, encompassing stop word removal, lemmatization, and subsequent tokenization. This leads to the creation of a lexicon denoting word frequencies, which is then converted into a document-term matrix. Utilizing the “LdaModel” function from “gensim”, we constructed an LDA model targeting 10 topics. To optimize topic modeling, parameters such as “random state”, “chunk size”, “passes”, and “alpha” were adjusted. Through utilizing the LDA method for topic modeling on the ChatGPT-related tweet dataset, we elucidated the 10 most prominent “chatgpt” tweet topics. After listing the key terms representing these 10 topics, we further visualized the results using the “pyLDAvis” library, displaying the visualizations as screenshots within the Notebook (Figure 4).

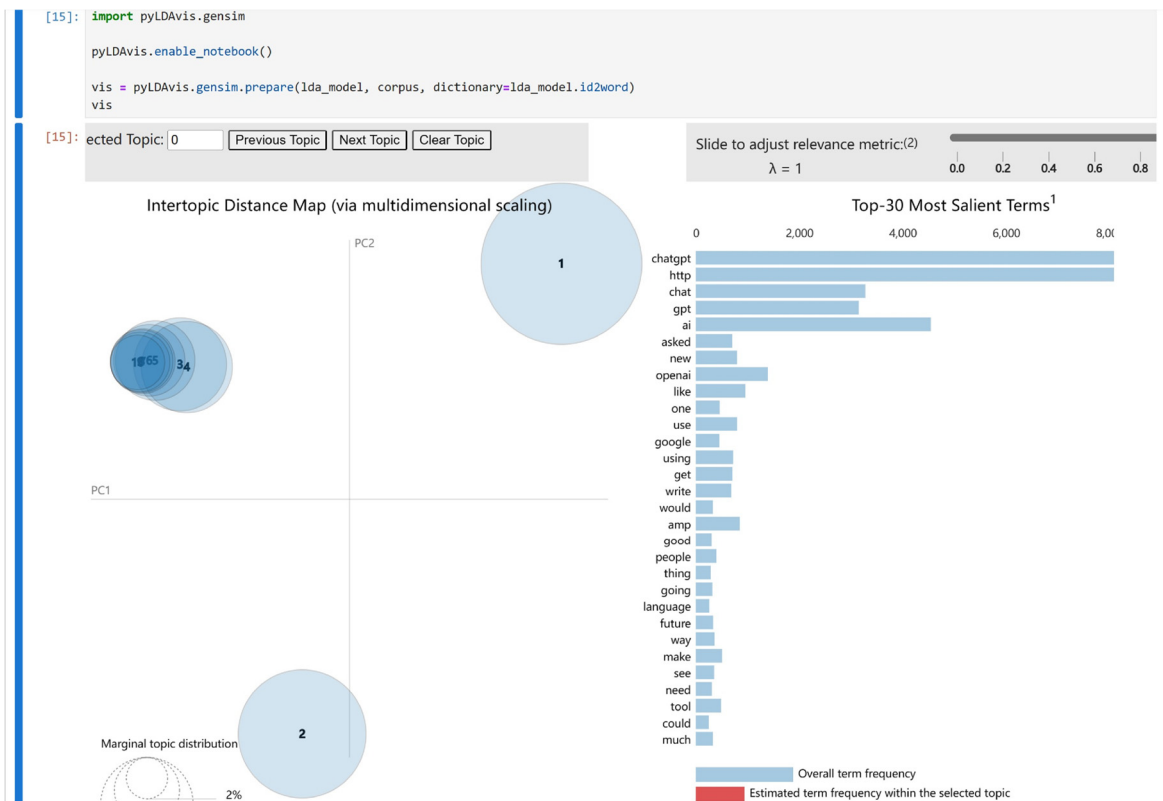


Figure 4. LDA topic modeling of tweets using the “gensim” and “pyLDAvis” libraries of Python.

3.3. Result and Discussion

3.3.1. Temporal Trend of Tweet Counts

In our analysis of tweet peaks:

- 7 February 2023: the unveiling of Google’s AI chatbot, Bard, marked its entry as a contender against ChatGPT;
- 8 February 2023: this date witnessed three significant developments:
 1. Alibaba Group announced its intent to develop an AI chatbot to challenge OpenAI’s ChatGPT;
 2. A study explored ChatGPT’s proficiency at generating academic content capable of evading anti-plagiarism tools;
 3. Another research introduced a framework for evaluating language learning models (LLMs) like ChatGPT, employing public datasets. ChatGPT’s performance on 23 datasets spanning eight distinct NLP tasks revealed strengths and limitations, including issues with reasoning accuracy and hallucinatory outputs.
- 15 March 2023: OpenAI made headlines with the release of its advanced language model, GPT-4;
- 17 March 2023: following the GPT-4 release, OpenAI’s CEO, Sam Altman, appeared on ABC News, outlining AI’s transformative potential and emphasizing the associated risks;
- 24 March 2023: OpenAI announced a feature for ChatGPT to integrate plugins, facilitating real-time information retrieval, calculations, and third-party interactions, all within a safety-centric framework.

On a weekly analysis:

The tweet volumes during the first three weeks of February were notably high, primarily driven by the events on 7 and 8 February. The bi-weekly count during the second and third weeks of March surpassed any other two-week segment in our observation

window, which is likely attributed to GPT-4's launch and the introduction of ChatGPT's plugin capabilities.

3.3.2. Analysis of Topic Classification and LDA Topic Modeling

Topic classification and LDA topic modeling are distinct processes. The built-in classifier categorizes individual tweets based on predefined topics. In this approach, each tweet is independently classified, and upon completion, tweets sharing the same topic are aggregated and ranked. Conversely, LDA topic modeling clusters the dataset holistically, generating multiple topics based on word probabilities.

This built-in classifier of Wolfram Mathematica assumes that the topic of the text is unique. The probabilities reflect the belief in these topics, not the proportion of topics. In Table 1, the first column presents the top 10 topics identified using the built-in classifier, ranked by the number of tweets they encompass. Despite the fact that LDA topic modeling is an unsupervised machine learning method and that it may not always provide coherent and understandable answers, we can attempt to interpret some of the topics. The second column of Table 1 presents descriptions of the six potential topics derived from our analysis of the LDA topic modeling.

Table 1. The comparison of topic classification and LDA topic modeling.

Topic Classification of Tweets with Built-in Classifier	LDA Topic Modeling of Tweets Using External Python Evaluation
1. Technology	• Seeking information or assistance, possibly from ChatGPT.
2. Video Games	• Engaging in dialogues concerning OpenAI's ChatGPT model, its functionalities, and integration with APIs.
3. Social Media	• The contribution of corporations like Microsoft to the advancement of AI and extensive modeling.
4. Career and Money	• Evaluating language models, their caliber, and the latest innovations in the domain.
5. Politics	• Exploring features of ChatGPT and the capacity of GPT in crafting text, including poetry.
6. School and University	• The application of AI technologies in professional settings and their prospective influence on activities such as content creation.
7. Books	
8. Music	
9. Health	
10. Movies	

4. Topic Classification of Tweets through the Integrative Use of GloVe, LDA, and KNN

In the pertinent literature concerning LDA research, various methodologies to extract pivotal keywords and implement text clustering have been proposed by numerous researchers. Chen et al. [26] introduced a novel model Multilevel Attribute Embedding for Semi-supervised User Identity Linkage (MAUIL), aiming to identify common user identities across various social networks. Within this model, LDA is employed to model user attributes at the topic level in conjunction with two other levels, namely the character and word levels, thereby facilitating a comprehensive framework for user identity mapping in a semi-supervised learning context. Kim et al. [27] proposed a technique that exploits the LDA model for keyword extraction and calculates the term frequency-inverse document frequency (TF-IDF) values, further applying them to the K-means clustering algorithm to acquire topics and analogous text content. A subsequent approach proposed by Wang et al. [28] leveraged an LDA-based text clustering algorithm, which synthesizes TF-IDF and LDA modeling, facilitating text clustering through the amalgamation of similarities. The notable sparsity of the matrix constructed via TF-IDF has been highlighted. A different avenue was explored by Kim et al. [29], who presented a model grounded in Word2Vec which, when integrated with K-means clustering, enhanced the ability to capture and represent corpus text aptly. Zheng [30] employed the LDA topic model to extract feature words and underlying topics, utilized Word2Vec to attain feature word vectors, and integrated both to realize text clustering. Shortly after the introduction of Word2Vec, Pennington et al. (2014) proposed the global vectors for word representation (GloVe) model, meticulously considering statistical information within the corpus, thereby enabling it to convey augmented semantic information. Li et al. [31] introduced the GV-LDA model

which, prior to LDA modeling, employs GloVe modeling for word vector extraction and substitutes words of high similarity to mitigate sparsity.

In summary, traditional clustering methodologies, when extracting features, model within a localized contextual window, often neglecting partial statistical information within the text collection. Achieving meticulous topic classification within a corpus poses inherent challenges, especially when the corpus pertains to specific details, such as tweets related to ChatGPT. In this chapter, a synergistic methodology entwining LDA, GloVe embeddings (Pre-trained model: “GloVe $\times \times \times$ -Dimensional Word Vectors Trained on Tweets”), and K-Nearest Neighbors (KNN) clustering is proposed to categorize topics within ChatGPT-related tweets. This comprehensive strategy ensures semantic, syntactic, and topical congruence within classified groups by utilizing the strengths of probabilistic modeling, semantic embeddings, and similarity-based clustering.

4.1. LDA and Jensen–Shannon Distance Calculation

In this section, a comprehensive methodology for utilizing the LDA model for topic representation, accompanied by an optimized approach for choosing the number of topics (K) and a metric for calculating text similarity using the Jensen–Shannon (JS) distance, is introduced.

Step 1: Initialization and Parameter Determination

Given a corpus D_{test} of preprocessed ChatGPT-related tweets containing N documents $\{1, 2, \dots, \} \{d_1, d_2, \dots, d_N\}$ and a vocabulary size of V , three primary hyperparameters must be defined for LDA modeling:

- α and β : Dirichlet prior parameters, typically assumed to be symmetric and utilized in default values;
- K : the number of topics to be identified.

In this study, while α and β are maintained at their default values, an exhaustive exploration is performed to find the optimum value for K , leveraging the perplexity metric:

$$\text{Perplexity}(D_{\text{test}}) = \exp \left\{ - \frac{\sum_{d=1}^N \sum_{i=1}^{N_d} \log P(w_d)}{\sum_{d=1}^N N_d} \right\}$$

where D_{test} , N , N_d , and $P(w_d)$ denote the test set, number of documents, word number in document d , and word probability, respectively.

Step 2: Gibbs Sampling for Distribution Estimation

Gibbs sampling is utilized to approximate the posterior distribution $P(Z|W, \alpha, \beta)$ and subsequently estimate the document–topic and topic–word distributions (θ and ϕ , respectively). The primary objective is to maximize the joint probability:

$$P(\theta, \phi, Z, W | \alpha, \beta) = P(\theta | \alpha) P(\phi | \beta) P(Z | \theta) P(W | Z, \phi)$$

where θ , ϕ , Z , and W represent the document–topic distribution, topic–word distribution, topic assignments, and words, respectively. The iterative sampling process continues until convergence.

Step 3: Topic Number Optimization using Perplexity

A search across a plausible range for K is conducted to locate the value that minimizes perplexity, assuring that the chosen number of topics provides the most coherent and distinctive categorization with minimal information loss.

Step 4: Calculating Text Similarity using the Jensen–Shannon Distance

For topic distributions P and Q retrieved from the LDA model, the JS distance is calculated to measure similarity:

$$D_{\text{JS}}(P \parallel Q) = \frac{1}{2} D_{\text{KL}}(P \parallel M) + \frac{1}{2} D_{\text{KL}}(Q \parallel M)$$

with $M = \frac{1}{2}(P + Q)$, and the Kullback–Leibler divergence defined as:

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

This step facilitates the quantification of the semantic similarity between pairs of documents, paving the way for efficacious clustering in subsequent stages. This detailed LDA methodology ensures a meticulous representation of topics within ChatGPT-related tweets, incorporating a meticulous parameter optimization step and a robust similarity metric. The consequent topic representations will feed into the integrative clustering strategy outlined in the forthcoming sections, guaranteeing the synthesis of coherent and semantically congruent clusters.

4.2. GloVe Embedding and Cosine Similarity Calculation

In this section, we meticulously delve into the methodology for implementing GloVe embedding and calculating cosine similarity within the purview of ChatGPT-related tweets. Importantly, we employ the pre-trained “GloVe $\times \times \times$ -Dimensional Word Vectors Trained on Tweets” model to convert each word in the tweets into a respective vector.

Step 1: GloVe Embedding

By leveraging the GloVe model, which is pre-trained on Twitter data, word embeddings captivate the semantic relationships among words, thereby playing a pivotal role in comprehending the contextual meaning nestled in tweets.

Given a word w_i in tweet t_j , the word is mapped to a vector using the GloVe model:

$$v(w_i) = \text{GloVe}(w_i)$$

Step 2: Weighting Scheme

Upon procuring the word vectors, they are accorded a weight based on the pertinence of the words to ChatGPT. The weighting function ω might be computed by leveraging the term frequency-inverse document frequency (TF-IDF) or another pertinent metric:

$$\omega(w_i) = f(w_i, \text{ChatGPT})$$

The resultant vector $v'(w_i)$ for word w_i is computed as follows:

$$v'(w_i) = \omega(w_i) \cdot v(w_i)$$

Step 3: Tweet Vector Representation

Assuming tweet t_j is constituted of N words, the vector representation $V(t_j)$ of the tweet is computed by amalgamating the vectors of individual words:

$$V(t_j) = \frac{1}{N} \sum_{i=1}^N v'(w_i)$$

Step 4: Cosine Similarity Calculation

To calculate the similarity between two tweets, specifically t_j and t_k , given their respective vector representations $V(t_j)$ and $V(t_k)$, cosine similarity is defined as follows:

$$\text{sim}(t_j, t_k) = \frac{V(t_j) \cdot V(t_k)}{|V(t_j)| |V(t_k)|}$$

where:

- The dot symbol (\cdot) represents the dot product of two vectors

- $|V(t_j)|$ and $|V(t_k)|$ represent the magnitude of the vectors $V(t_j)$ and $V(t_k)$, respectively.

$$|V(t_j)| \text{ and } |V(t_k)|$$

This cosine similarity value ranging from -1 to 1 insinuates that a score of 1 implies that the two tweets are identical, -1 implies that they are diametrically opposed, and a value approximating 0 suggests negligible similarity. This measure, with its robustness for high-dimensional spaces, has witnessed widespread adoption for text similarity computations.

In the ensuing phase, this computed similarity measure will be employed to group similar tweets via KNN clustering. Engaging GloVe embeddings that are specifically trained on tweets is anticipated to augment the relevancy and precision of the similarity computations, thereby fostering a more contextually pertinent cluster formation in the successive steps.

4.3. Combining Similarity Metrics and KNN Clustering

In this section, the concentration is pivoted towards delineating the process of amalgamating the two aforementioned similarity metrics—deriving from LDA (calculated with Jensen–Shannon distance) and GloVe embeddings (calculated with cosine similarity)—and utilizing the unified metric for the KNN clustering of the ChatGPT-related tweets.

Step 1: Combining Similarity Metrics

Let us denote the JS distance between tweet t_i and t_j as $S(t_i, t_j)$ and the cosine similarity as $COS(t_i, t_j)$. An aggregated similarity score, denoted as $S(t_i, t_j)$, is defined as follows:

$$S(t_i, t_j) = \alpha \cdot JS(t_i, t_j) + (1 - \alpha) \cdot COS(t_i, t_j)$$

Here, α is a hyperparameter, $0 \leq \alpha \leq 1$, which adjusts the influence of each similarity metric on the final combined similarity score. This hyperparameter can be optimized using a grid search or similar parameter optimization technique by assessing the quality of resulting clusters through pertinent evaluation metrics, such as a silhouette score or the Davies–Bouldin index.

Step 2: KNN Clustering

Employing $S(t_i, t_j)$ as the similarity measure, KNN clustering is employed to group tweets into respective thematic clusters. The algorithm can be explicated as follows:

1. Initialization: select k initial centroids, where k is the predetermined number of clusters;
2. Assignment step: assign each tweet t_i to the cluster C_j with the nearest centroid, employing $S(t_i, t_j)$ to compute similarity/distance:

$$C_j = \operatorname{argmin}_{c \in C} S(t_i, c)$$

3. Update step: update the centroids by computing the mean vector of all tweet vectors within each cluster;
4. Convergence check: evaluate if the centroids have shifted. If the centroids remain unaltered or the shift is below a predetermined threshold, proceed to the next step, else return to the assignment step;
5. Result retrieval: final clusters C are obtained and can be further analyzed for topical insights.

It is imperative to highlight that the value of k (number of clusters) significantly influences the ensuing clustering results. Various methods, such as the elbow method or silhouette analysis, can be deployed to ascertain an optimal k value, thus ensuring that the clusters are neither too sparse nor too dense.

4.4. Experiments and Discussion

4.4.1. Data Source

The dataset employed in this study is derived from “Task A of SemEval-2016 Task 6” [32]. The “tweet” field from each record within the dataset was extracted to formulate the test dataset required for this paper. The dataset encompasses over 2800 tweets, categorized into the following five thematic classes: “Legalization of Abortion” (603 tweets), “Climate Change is a Real Concern” (496 tweets), “Feminist Movement” (646 tweets), “Hillary Clinton” (639 tweets), and “Atheism” (514 tweets).

4.4.2. Perplexity Test

The perplexity value was utilized to determine the number of topics, denoted as K . Experiments were conducted 10 times, with the perplexity corresponding to different K values being the average of the results from the 10 experiments, as illustrated in Figure 5. The experimental results from Figure 5 indicate that when K equals five, the perplexity is minimized, suggesting optimal LDA modeling at this point; hence the optimal number of topics, K , is five.

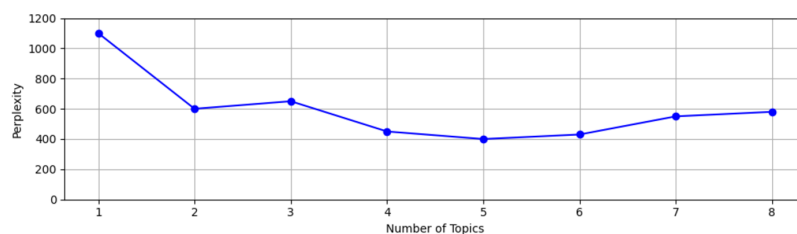


Figure 5. Perplexity value of the LDA model under different numbers of topics.

4.4.3. Algorithm Validation

To validate the efficacy of the proposed tweet clustering algorithm, which integrates LDA, GloVe, and KNN, it was compared with models based on LDA, LDA + TF-IDF, and LDA + Word2Vec, all within the KNN clustering algorithm. Comparative metrics included accuracy, precision, recall, and F1 scores. The comparative scenarios of accuracy, precision, recall, and F1 score for the five topics are depicted in Figures 6–9, respectively. Figure 10 provides a synthesis of the aforementioned experimental results. As can be discerned from Figure 10, the results of the four clustering algorithms gradually ascend in terms of accuracy, precision, recall, and F1 scores, with the clustering algorithm proposed in this paper, which amalgamates the LDA and GloVe models, demonstrating optimal performance across all four indicators. This can be attributed to the algorithm’s fusion of topic and word distributions as extracted by the LDA model and combined with the GloVe pre-trained model, which considers the characteristics of global semantic information and utilizes statistical information from text aggregation, thereby enhancing the clustering effect of tweet topic classification.

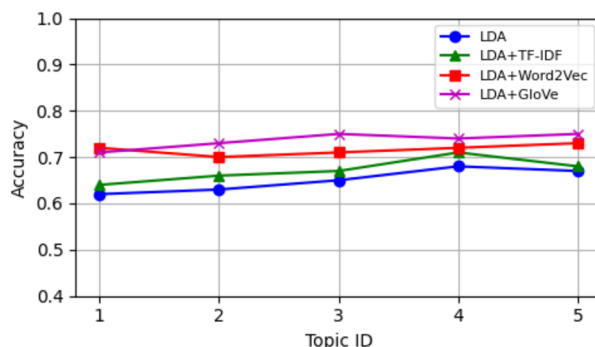


Figure 6. Comparison results of accuracy under different topics.

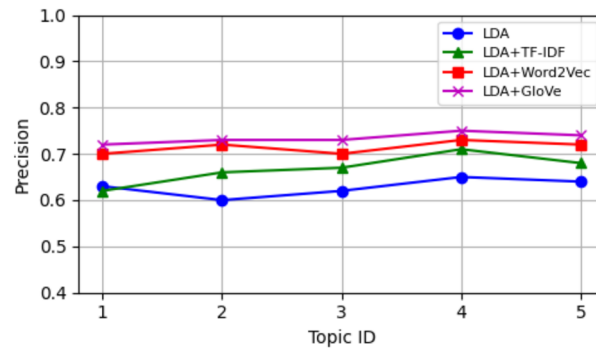


Figure 7. Comparison results of precision under different topics.

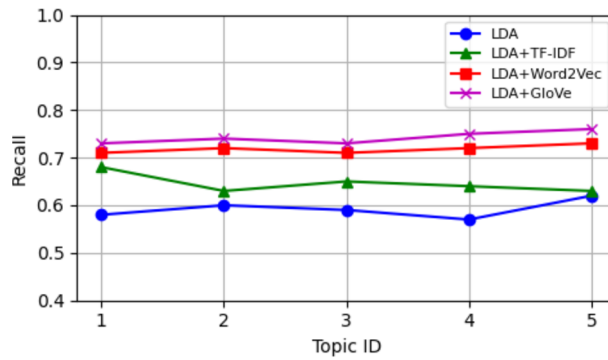


Figure 8. Comparison results of recall under different topics.

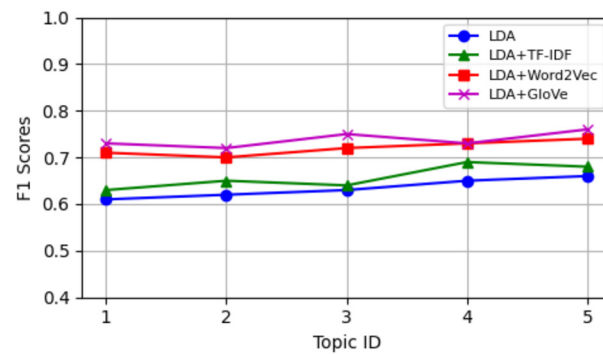


Figure 9. Comparison results of F1 scores under different topics.

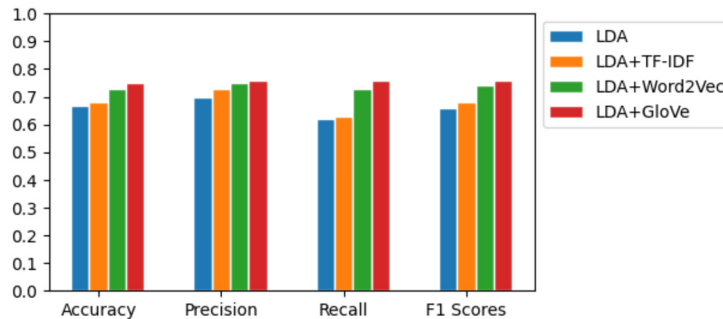


Figure 10. Comparison of accuracy, precision, recall rate, and F1 scores.

5. Tweets Sentiment Analysis Based on Transfer Learning

5.1. Research Questions

- RQ4. How effective is the use of Wolfram Mathematica’s built-in classifier for the sentiment analysis of tweets?

- RQ5. How can pre-trained models in the Wolfram Neural Net Repository be leveraged for transfer learning to perform sentiment analysis on tweets?

5.2. Method

5.2.1. Data Source

In this chapter, we use “the sentiment140 dataset” that has been extensively applied in related Twitter sentiment classification studies [33]. The dataset contains 1,600,000 tweets extracted using the Twitter API. The tweets have been annotated, and they can be used to detect sentiment. The dataset contains the following six fields:

target: the polarity of the tweet (0 = negative, 4 = positive);

ids: the id of the tweet (2087);

date: the date of the tweet (Sat 16 May 23:58:44 UTC 2009);

flag: the query (lyx). If there is no query, then this value is NO_QUERY;

user: the user that tweeted (robotickilldozr);

text: the text of the tweet (Lyx is cool).

In consideration of computational resource constraints, a random subset of 100,000 records was selected from the dataset for this study. Of these, 80,000 records were allocated to the training set, while the remaining 20,000 records constituted the validation set.

5.2.2. Base Line: Built-in Classifier

In addressing RQ4, we employed two distinct methods using the built-in classifier to conduct a sentiment analysis of tweets. Through empirical investigation, we discerned their efficacy, which is subsequently established as the baseline for sentiment analysis research in this paper.

“Classify” Function—“Sentiment” option

Wolfram Mathematica already has a built-in classifier for text sentiment analysis. The classifier assumes the text conveys only one sentiment. The probabilities reflect the belief in these sentiments, not the proportion of sentiments. The current version only works for the English language. Although the built-in classifier is relatively convenient to use, its performance in the sentiment analysis of tweets is not satisfactory.

The primary reason for this unsatisfactory performance may be that the built-in classifier is designed for general text sentiment analysis rather than specifically for tweets, which have their own unique format and user expression patterns. Additionally, the built-in sentiment classifier categorizes sentiment into three classes (positive, neutral, negative), whereas our dataset only contains two sentiment labels (positive, negative). This discrepancy may introduce some bias when evaluating the performance of the built-in sentiment classifier. There are many functions available for natural language processing in Wolfram Mathematica, which allow for the creation of neural network create some experiments and tests to compare methods and choose the best result.

“Classify” Function—“Machine Learning” option

Furthermore, we can also employ the “Classify” function, utilizing its inherent machine learning methodologies (“DecisionTree”, “LogisticRegression”, “Markov”, “Naïve-Bayes”, “SupportVectorMachine”, “NeuralNetwork”) to train a sentiment classifier for tweets. This will serve as a baseline for our research. Subsequently, we employed the “Markov” method to train a sentiment classifier (Figure 11).

5.2.3. Comparative Analysis with Existing Sentiment Analysis Tools

In the domain of sentiment analysis, approaches predominantly fall under two categories: machine learning-based and lexical-based methods. Machine learning methodologies predominantly employ supervised classification techniques, usually categorizing sentiments as either positive or negative. These methods necessitate the availability of labeled data to train the classifiers, presenting a notable challenge due to the scarcity of such data, as articulated by Pennebaker, Francis, and Booth [34]. The dearth of labeled data restricts the efficacy and applicability of machine learning methods to novel datasets, a

constraint often attributed to the intensive resources required to label data, even for tasks of minimal complexity.

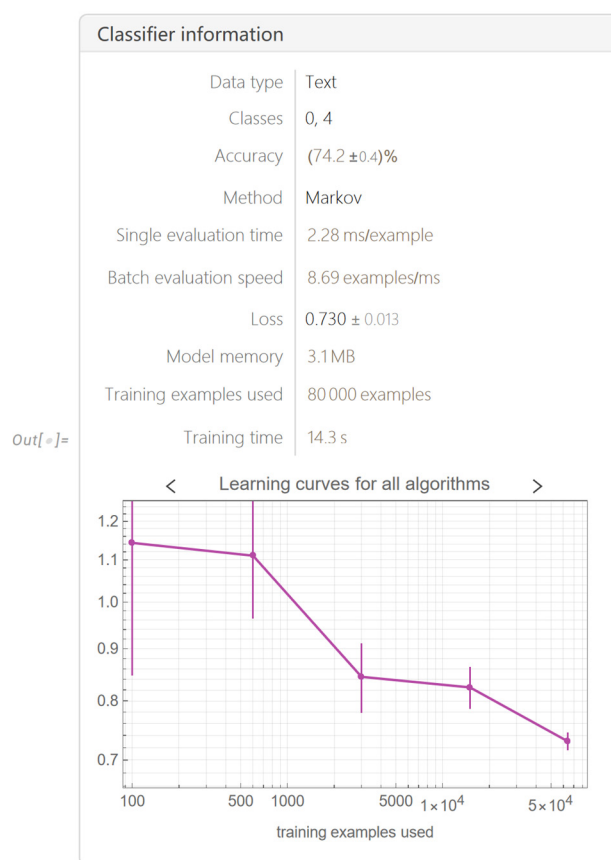


Figure 11. Training a sentiment analysis classifier for tweets using the “Classify” function’s ML capabilities.

Conversely, lexical-based methodologies operate on predefined word lists, associating individual words with specified sentiments and polarities [35]. These methods exhibit variability and are contingent upon the contexts of their creation. Although they circumvent the need for labeled data, the obstacle inherent to lexical-based methods is the necessity to identify unique lexical dictionaries versatile enough to be adapted to diverse contexts [36]. This is particularly relevant in the realm of microblogging, where conventional lexical dictionaries often fail to encompass slangs, abbreviations, and internet acronyms prevalent in platforms like Twitter due to character limitations.

In a more contemporary study conducted by Bonta and Janardhan [37], an analytical comparison was executed to evaluate the efficacy of various lexical methods, namely NLTK, VADER, and TextBlob, in accurately classifying sentiments derived from film reviews. The empirical analysis was based on a compilation of 11,861 sentence-level excerpts acquired from www.rottentomatoes.com, serving as the evaluative framework for assessing the operational proficiency of the mentioned lexicons. The findings from their rigorous examination elucidated that VADER demonstrated superior performance in sentiment categorization compared with its counterparts. The results imply a noteworthy difference in the ability of these lexicons to accurately discern and categorize sentiments in the context of movie evaluations, with VADER emerging as the most proficient among the evaluated methods.

While lexical-based methods and those leveraging tools or libraries in Python or R languages have their respective merits, it has been observed that in the domain of textual sentiment analysis, methods anchored in deep learning have witnessed accelerated advancements and are exhibiting more promising prospects in recent times. Subsequently,

a comparison between deep learning methodologies and existing tools or libraries is presented below:

Learning methodology and complexity: Deep learning models, such as LSTM, BERT, and transformers, learn complex representations from data, enabling them to understand contextual information, semantic relationships, and nuances in texts, which are critical for accurate sentiment analyses. NLTK, TextBlob, and VADER generally rely on simpler, rule-based, or lexicon-based methods, possibly leading to lower accuracy, especially in understanding context and nuances in sentiments.

Adaptability and generalization: Deep learning models can adapt to various domains and are versatile, given sufficient and relevant training data. They can generalize well across diverse text types, languages, and domains, providing more robust solutions for sentiment analysis. Existing tools have fixed rules or lexicons, limiting their adaptability to new domains or text types. Their generalization capacity is often restricted.

Resource consumption and scalability: Deep learning models often require substantial computational resources, time, and large labeled datasets for training, making them resource intensive. Existing tools like NLTK, TextBlob, and VADER are lightweight, less resource-intensive, and can be effective when computational resources are limited or when working with small datasets.

Interpretability and ease of use: Deep learning models are often considered as “black boxes” due to their complexity, making interpretability a challenge. They also necessitate expertise in machine learning and deep learning frameworks. Existing tools are generally more interpretable and user friendly, suitable for users with limited knowledge in machine learning and offering clear insights into how sentiment scores are derived.

Performance and accuracy: Deep learning models, with their advanced capability to learn intricate patterns and representations, tend to provide superior accuracy and performance in sentiment analysis tasks across diverse and large-scale datasets. Existing tools, being less sophisticated, may struggle with accuracy, particularly in handling ambiguous, sarcastic, or contextually rich texts.

While deep learning models exhibit pronounced advantages in accuracy, adaptability, and applicability in diverse sentiment analysis tasks, their resource consumption, complexity, and interpretability pose significant challenges. Conversely, existing tools like NLTK, TextBlob, and VADER, with their simplicity, interpretability, and less resource-intensive nature, offer practical solutions for quick or preliminary analyses, especially in resource-constrained scenarios. Each approach has its merits and limitations, and the choice between them should be guided by the specific requirements, constraints, and goals of the sentiment analysis task at hand.

5.2.4. Transfer Learning Based on Wolfram Neural Net Repository

Most of the existing methods of Twitter sentiment classification follow the method proposed by Pang et al. [38] and apply machine learning algorithms to build a classifier from tweets with a manually annotated sentiment polarity label. In recent years, there has been a growing interest in using deep learning techniques, which can enhance classification accuracy. Some conventional ML methods cost significant time and labor on task-specific feature engineering. In contrast, DL methods can automatically extract features using unsupervised or semi-supervised learning algorithms [39]. Moreover, it can generate high-quality vector representations that differ from the low-quality vector representations generated by feature engineering [40]. However, the application of deep learning methods needs a large amount of annotated data. In some domains, it is challenging to construct a large-scale annotated dataset because of the costly expense of data acquisition and annotation.

Transfer learning can solve the problem by leveraging knowledge obtained from a large-scale source domain to improve the classification performance in the target domain [41]. At its simplest, migrating pre-trained word vectors initializes the input of the deep learning model. The pre-trained word vectors obtained based on massive text data are

an essential part of the learned semantic knowledge that can significantly improve natural language processing tasks based on deep learning. In natural language processing (NLP) tasks, there are several ways to employ transfer learning strategies. Generally, we can initialize input words by transferring pre-trained word embedding. The pre-trained word embeddings on a large-scale corpus contain abundant syntactic and semantic knowledge, which significantly promotes the NLP tasks based on deep learning methods [42].

The static word embeddings are 3 million 300-dimension Word2Vec word embeddings trained on GoogleNews, 1 million 300-dimension FastText word embeddings trained on Wikipedia, and 1.2 million 200-dimension GloVe word embeddings trained on Twitter. If the word is included in the pre-trained embedding, we can obtain the word vector directly. If not, we generate the word vector randomly. However, static word vectors such as Word2Vec only produce a fixed vector representation. They cannot solve the problem that the same word may have different meanings when it appears in different positions in the text. The emergence of deep neural networks allows language models to dynamically generate word vectors to solve the ambiguity of words in different situations. With the emergence of pre-trained language models such as bidirectional encoder representations from transformers (BERT) [43], the model can generate dynamic word embeddings to tackle polysemy. Recently, fine-tuning the pre-trained language model with limited, annotated domain-specific data has achieved excellent performance in a series of NLP tasks [44,45]. Deep contextualized word embeddings supported by the language model ELMo improve word representation quality and handle the polysemy problem to a certain extent. Different from the static word embeddings, it represents a word according to its context.

To find a transfer learning system that is able to extract comprehensive public sentiment on ChatGPT on Twitter with satisfying performance, four transfer learning approaches based on Wolfram Mathematica that pre-train word embedding models were proposed. More specifically, two of the methods involved separately transferring diverse word embeddings (“GloVe 100-Dimensional Word Vectors Trained on Tweets” and “ConceptNet Numberbatch Word Vectors V17.06”) and then processing them in an LSTM layer. The other two methods involved separately transferring diverse word embeddings (“BERT Trained on BookCorpus and English Wikipedia Data” and “ELMo Contextual Word Representations Trained on 1B Word Benchmark”) and then processing them in a linear layer and aggregation layer. The research presented in this section also addresses RQ5 of this paper. The following will discuss the four transfer learning models proposed in this paper in detail.

- GloVe + LSTM

Pre-trained model: GloVe $\times \times \times$ -Dimensional Word Vectors Trained on Tweets

In light of the above, our approach employs transfer learning by leveraging GloVe word embeddings pre-trained on Twitter data, owing to its proven efficacy at capturing semantic word relationships and contextual information. Transfer learning allows the model to utilize the knowledge acquired from vast datasets, aiding in superior model generalization, especially in scenarios with limited labeled data.

GloVe, or “Global Vectors”, is a word embedding technique that represents words in a high-dimensional space where the semantic relationship between the words is encoded by the distance between them. Chosen for its robustness in handling Twitter’s diverse linguistic constructs, GloVe embeddings serve as the input layer to our model, converting tokenized words into dense vectors that encapsulate semantic information. This Wolfram pre-trained model represents words as vectors. Released in 2014 by the computer science department at Stanford University, this 25/50/100/200-dimensional representation is trained on tweets using an original GloVe method. It encodes 1,193,515 tokens as unique vectors, with all tokens outside the vocabulary encoded as the zero vector. The token case is ignored. The training dataset consists of 27 billion tweets, with around 1.2 million unique tokens. All tokens are uncased.

Subsequently, long short-term memory (LSTM) networks, a variant of recurrent neural networks (RNNs), are used to process the sequential data. LSTMs are particularly well suited

for tasks involving sequences due to their ability to remember long-term dependencies in the data, mitigating the vanishing gradient problem common in traditional RNNs. In our architecture, LSTMs aid in capturing the sequential information present in the text data, processing the embedded word vectors from GloVe to analyze the sentiment of the text.

The amalgamation of GloVe embeddings and LSTM forms our model’s backbone. Below is the detailed architecture:

1. GloVe embedding layer: converts tokens to dense vectors representing words;
2. Dropout layer: mitigates overfitting by randomly setting a fraction of input units to 0 at each update during training;
3. LSTM layer: processes sequential data, capturing long-term dependencies between words;
4. Sequence last layer: extracts the relevant features from the output sequence;
5. Linear layer: performs linear transformations to the extracted features;
6. Softmax layer: converts model outputs to probability distributions for each class.

The training of our model involved the use of categorical cross-entropy loss and the Adam optimization algorithm, with learning rate annealing and early stopping being employed as regularization techniques. The model training results are also illustrated in Figure 12.

- ConceptNet + LSTM

Pre-trained model: ConceptNet Numberbatch Word Vectors (V17.06)

Our study also employs another transfer learning approach, “ConceptNet + LSTM,” leveraging the pre-trained ConceptNet Numberbatch Word Vectors (V17.06) from the Wolfram Neural Net Repository. ConceptNet Numberbatch, released in 2016, represents words as high-dimensional vectors, fusing human-constructed knowledge from the ConceptNet graph with multiple distributional-based embeddings like GloVe, Word2Vec, and FastText, which were trained on the Open Subtitles 2016 dataset.

ConceptNet Numberbatch was chosen for its comprehensive representation of words, which combines semantic, grammatical, and factual knowledge, allowing for a more enriched representation of words in textual data, which is particularly beneficial for analyzing the intricate and multifaceted nature of sentiments in text.



Figure 12. The architecture of the “GloVe + LSTM” deep neural network and its training results.

In this model, each word is mapped to a unique vector in a high-dimensional space, enabling the encapsulation of semantic meanings and relationships between words. Any tokens that are not part of the 400,000+ unique tokens in the pre-trained model's vocabulary are represented as a zero vector. The original model's tokens, which had underscores, have been modified, replacing underscores with white spaces.

Our "ConceptNet + LSTM" model's architecture, depicted in Figure 13, is structured with sequential layers, each serving a distinct purpose in processing and transforming the input data.

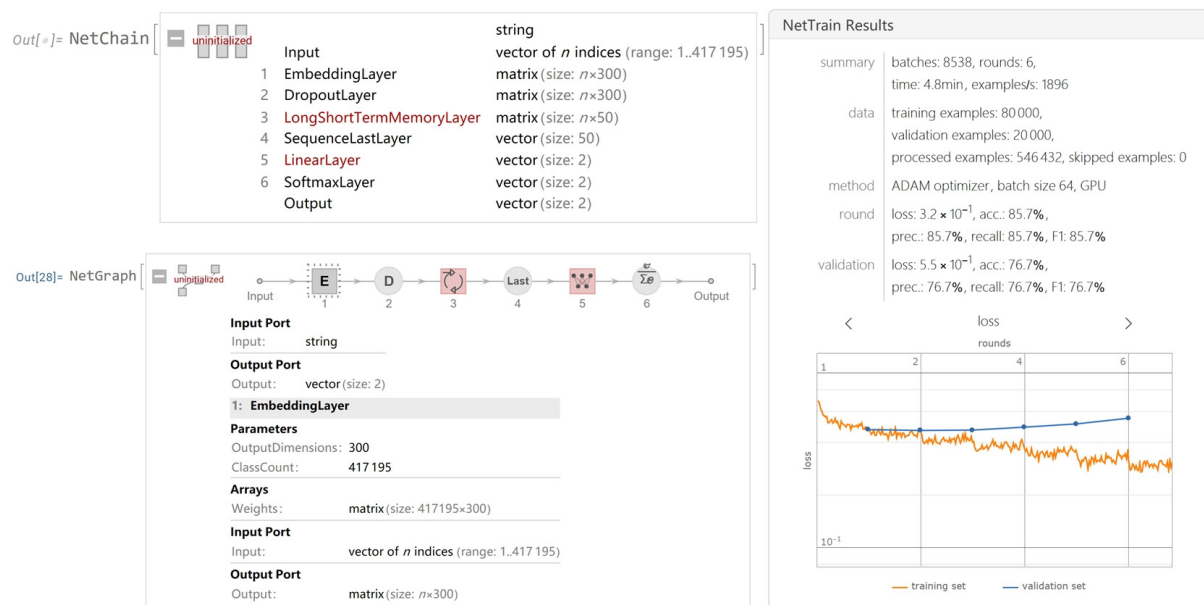


Figure 13. The architecture of the "ConceptNet + LSTM" deep neural network and its training results.

1. ConceptNet embedding layer: transforms tokens to dense vectors using the enriched representations from ConceptNet;
2. Dropout layer: acts as a regularization mechanism, reducing the likelihood of overfitting by randomly zeroing a fraction of the input units during training;
3. LSTM layer: addresses the sequence processing, making sense of the order and context of words in the text;
4. Sequence last layer: extracts the significant features from the sequence produced by the LSTM layer;
5. Linear layer: applies linear transformations to the features for mapping to the output space;
6. Softmax layer: converts the model's raw outputs into probabilities, indicating the likelihood of each class.

Figure 13 also illustrates the results attained from the training process, shedding light on the model's learning trajectory and its ability to make sense of the sentimental nuances in the text data.

- Fine-Tuning BERT

Pre-trained model: BERT trained on BookCorpus and English Wikipedia data

Our research also incorporates the advanced technique of fine-tuning BERT models, using the pre-trained BERT model from the Wolfram Neural Net Repository, trained on BookCorpus and English Wikipedia data. BERT, or bidirectional encoder representations from transformers, introduced in 2018, revolutionized the representation of text by learning deep bidirectional contextual representations.

BERT was chosen for its unparalleled ability to capture context on both sides of a word (left and right), allowing for a richer and more accurate representation of text semantics. It is

known for setting state-of-the-art performance across various NLP tasks. BERT operates on the principle of bidirectional self-attention or transformer encoder, learning representations by jointly conditioning on both the left and right context in all layers. This approach allows for the creation of highly contextualized word representations.

In our “Fine-Tuning BERT” model, depicted in Figure 14, the architecture is composed of sequential components, each designed to accomplish specific tasks:

1. BERT embedding layer: transforms text into sequences of high-dimensional vectors utilizing the pre-trained BERT representations;
2. Dropout layer: implements regularization by randomly setting a subset of input units to zero at each update during training time to prevent overfitting;
3. Linear layer: applies linear transformation to the input data for mapping to the output space;
4. Aggregation layer: utilized to convert an array with spatial dimensions into a fixed-size vector representation, crucial for classifying sequences of subword embeddings using a max pooling strategy;
5. Softmax layer: normalizes the raw output values from the model, producing a probability distribution over the classes.

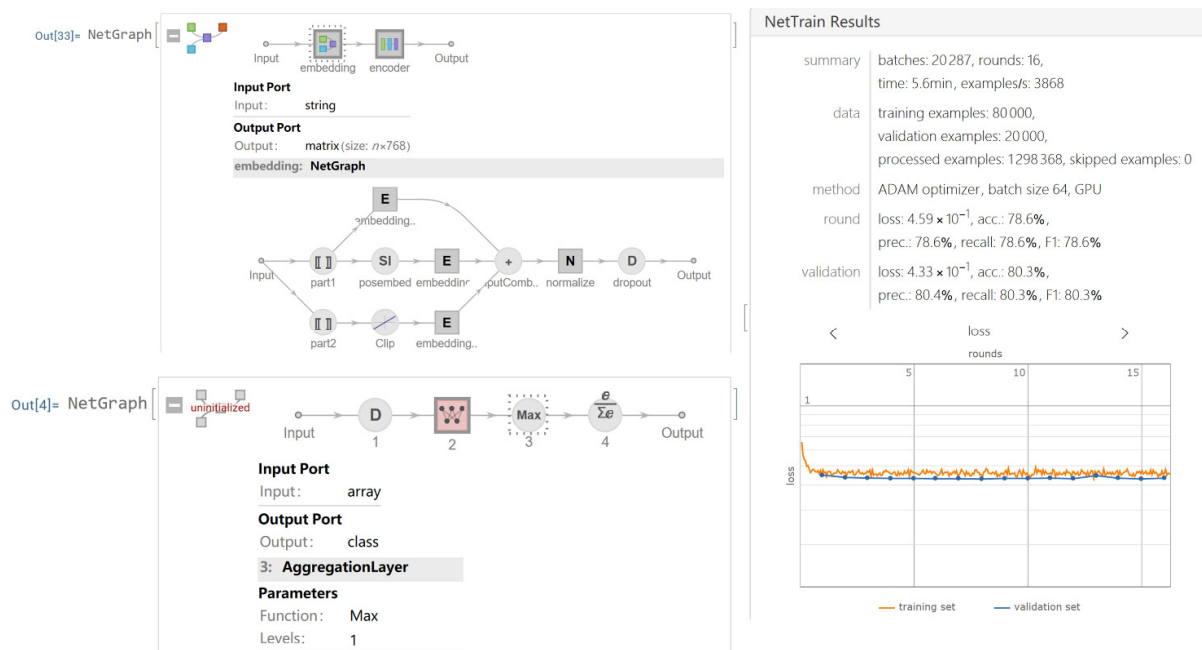


Figure 14. The architecture of the “ Fine-Tuning BERT “ deep neural network and its training results.

The training process and resulting model performance are illustrated in Figure 14, demonstrating the model’s proficiency in understanding and interpreting the subtleties of sentiment in textual data. Within Wolfram Mathematica, we employ an aggregation layer as the terminal stage in a series of convolutions, poolings, etc., enabling the transformation of a spatial array into a definitive vector representation. This architecture is vital for our “Fine-Tuning BERT” model, and it is designed to categorize sequences of sub-word embeddings effectively.

- Fine-Tuning ELMo

Pre-trained model: ELMo contextual word representations trained on 1B Word Benchmark

In addition to the previously mentioned models, our research also explores the power of “Fine-Tuning ELMo”, utilizing the pre-trained ELMo model from Wolfram trained on

the 1B Word Benchmark. ELMo, or Embeddings from Language Models, is a revolutionary model introduced in 2018 by the Allen Institute for Artificial Intelligence (AI2).

ELMo was selected due to its innovative representation of words as contextual word embeddings. Unlike traditional methods, ELMo generates embeddings based on the entire sentence context, allowing for the representation of nuanced meanings and uses of words based on their surrounding context.

ELMo leverages a deep bidirectional language model to produce three vectors for each token; two are contextual and one is non-contextual. It captures the semantic essence of the word based on its occurrence in the sentence, rendering a richer and more robust representation of words, which are linearly combined and are character- and case-sensitive.

Figure 15 illustrates the architecture of our “Fine-Tuning ELMo” model, composed of the following sequential components:

1. ELMo embedding layer: utilizes the pre-trained ELMo representations to convert tokens into contextual word-embedding vectors;
2. Dropout layer: introduces regularization by randomly deactivating certain neurons during training to mitigate the risk of overfitting;
3. Linear layer: executes linear transformations to map the obtained features to the desired output space;
4. Aggregation layer: converts spatial arrays into a unified vector representation, crucial for the classification of sequences of subword embeddings using a max pooling strategy;
5. Softmax layer: normalizes the model outputs to represent probabilities, facilitating the identification of the most likely class.

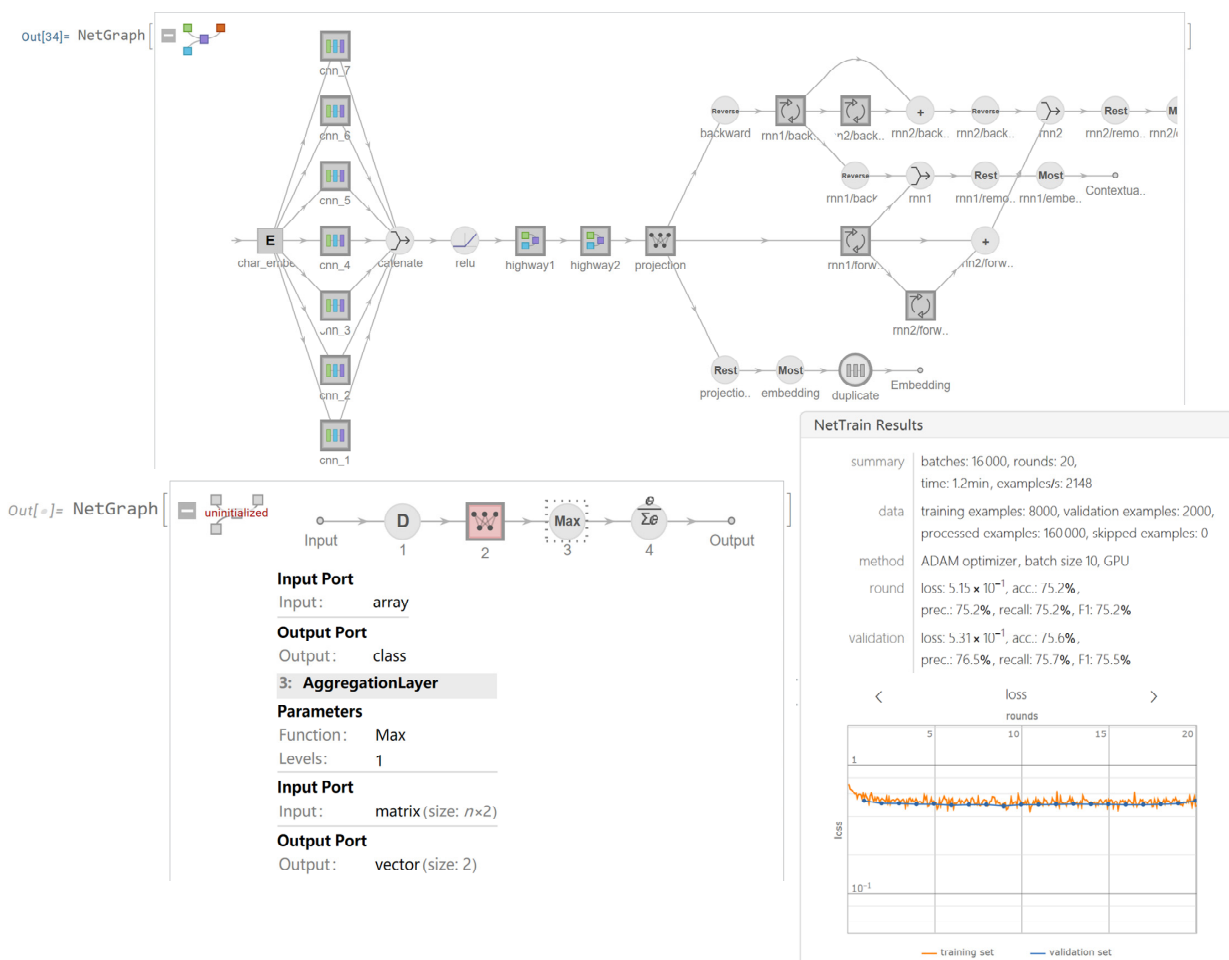


Figure 15. The architecture of the “Fine-Tuning ELMo” deep neural network and its training results.

Similar to the “Fine-Tuning BERT” model, the “Fine-Tuning ELMo” model is meticulously fine-tuned and optimized using the categorical cross-entropy loss and the Adam optimizer, aiming to achieve superior performance in sentiment analysis tasks. The model’s learning and its adeptness at extracting sentimental nuances from the textual data are depicted in Figure 15.

ELMo’s uniqueness lies in its ability to construct word representations based on characters, making it independent of a fixed vocabulary. This capability enables ELMo to interpret and generate representations for any word, irrespective of its presence in the training data, thus allowing for a more versatile and comprehensive understanding of textual data.

5.2.5. Sentiment Analysis of the Tweets Dataset Related to ChatGPT

In this section, we delve into the sentiment analysis of the top 10,000 tweets related to ChatGPT, ranked by their respective number of likes. We employ two distinct methodologies for this analysis. First, we use the built-in sentiment classifier available in Wolfram Mathematica. Second, we utilize the “GloVe + LSTM” model, which was previously designed and trained. This model enables us to effectively analyze the sentiment of the tweet dataset related to ChatGPT in our study.

To elucidate the comparative performance of these approaches, we visualize the distribution of sentiment polarities as detected by each method across the dataset of 10,000 tweets. Additionally, we provide a temporal analysis of sentiment trends, aggregating data on a weekly basis. This analysis aims to uncover any temporal variations in sentiment polarity (Figure 16).

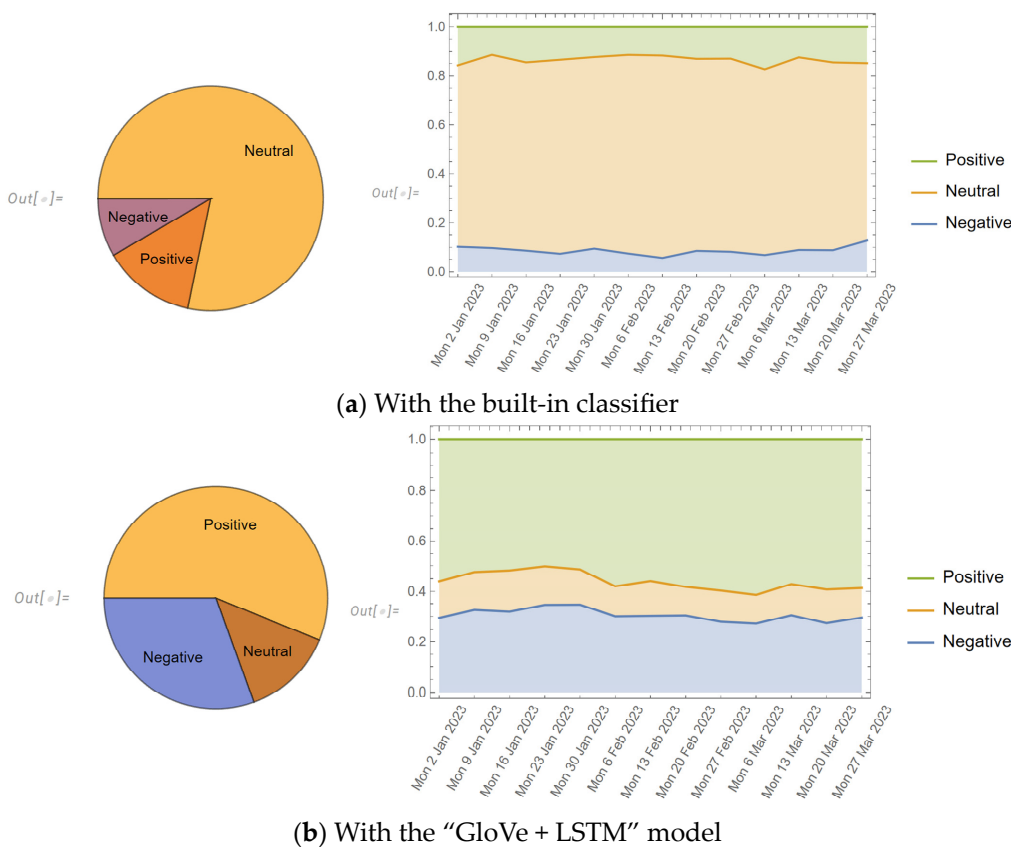


Figure 16. Proportional distribution and temporal trends of sentiment polarities in the top 10,000 liked tweets.

When utilizing the built-in classifier, a noticeably larger number of tweets were categorized as “Neutral” in terms of sentiment polarity. In contrast, when applying the

“GloVe + LSTM” model, there was a significant prevalence of tweets classified as “Positive.” Considering various media reports about ChatGPT, this outcome aligns well with our empirical observations and intuitive expectations. On the other hand, when examining the temporal trends of sentiment polarities, the public attitude towards ChatGPT appears consistent. However, there was a slight increase in the number of tweets classified as “positive” after February.

5.3. Result and Discussion

In this research, we introduced and examined four distinct transfer learning methodologies within the realm of deep learning. Transfer learning, as an essential paradigm, permits the deployment of knowledge from a previously learned task to a new yet similar task, obviating the need for extensive training data. To comprehensively assess the efficacies of these four methodologies, a series of experiments were conducted. The performance metrics employed for the general research evaluation encompass “Accuracy”, “Precision”, “Recall”, and “F1 Score”. Each of these metrics offers a unique perspective on the model’s capability; while “Accuracy” gauges the overall correctness of predictions, “Precision” and “Recall” provide insights into the model’s true positive rate and the ability to capture all positive instances, respectively. The “F1 Score”, being the harmonic mean of precision and recall, furnishes an aggregate measure of a model’s robustness. The subsequent sections delve into the intricacies of each method, providing a comparative analysis based on the four metrics of accuracy, precision, recall, and F1 Score (Table 2).

Table 2. Evaluation of the built-in classifier and four distinct transfer learning methodologies.

	Classify- “Sentiment”	Classify-ML- Markov	GloVe- LSTM	ConceptNet- LSTM	FT-BERT	FT-ELMo
Accuracy	56.0	74.2 ± 0.4	81.1	76.7	80.3	75.6
Precision	-	-	81.1	76.7	80.4	76.5
Recall	-	-	81.1	76.7	80.3	75.7
F1 Scores	-	-	81.1	76.7	80.3	75.5

We aim to identify a transfer learning system that can effectively extract comprehensive public sentiment on ChatGPT from Twitter with satisfactory performance. To harness syntax and semantics pre-trained on a vast corpus, two methods of transferring diverse word embeddings were integrated with an LSTM layer. As static word embeddings cannot address polysemy, we introduced two additional methods: fine-tuning BERT and fine-tuning ELMo. This approach allowed us to capitalize on the robust feature extraction capability of large neural networks by using minimal annotated target-domain data to fine-tune the language model.

Consistent with the existing literature, both BERT and ELMo are recognized as considerably large pre-trained models that convert individual words into high-dimensional vectors. BERT operates in a dimensionality of 768, while ELMo boasts a massive dimensionality of 3072 (1024×3). However, empirical findings suggest that, for the specific task of sentiment analysis on tweets, the “GloVe100 + LSTM” model outperformed the others. A plausible reason for this distinction might be the training corpus used for GloVe100, which was curated from Twitter datasets. In contrast, BERT and ELMo were trained on more comprehensive corpora that are not tailored for tweet-like structures.

For practical applications, the GloVe100 + LSTM model, owing to its utilization of 100-dimensional word vectors, demands fewer computational resources, resulting in shorter training durations. Conversely, both the Fine-BERT and Fine-ELMo models are resource-intensive, necessitating extended training phases. They are especially prone to computational constraints, potentially leading to premature termination, particularly when operated on standard personal computers. Thus, when employing the Wolfram Mathematica computing platform for deep learning instruction and related research, the “GloVe100 + LSTM” model emerges as a more practical and user-friendly alternative.

In our research, we have devised four distinct transfer learning models for the sentiment analysis of tweets related to ChatGPT. These models are constructed based on various pre-trained models from the Wolfram Neural Net Repository and are inherently generic, allowing them to be applied to sentiment analysis of tweets concerning diverse themes or subjects beyond those specifically related to ChatGPT. Thus, the models and methodologies proposed in our study are not confined to analyzing sentiments in ChatGPT-related tweets but extend to a broader range of social media data, providing a versatile framework for sentiment analysis across diverse topics and domains.

Among the proposed models, “Fine-Tuning BERT” and “Fine-Tuning ELMo” are particularly noteworthy. These models integrate pre-trained models trained on extensive corpora as the nucleus of deep networks. The models are subsequently trained on a relatively smaller corpus of subject-specific texts, i.e., ChatGPT-related tweets, to refine and optimize their performance for sentiment analysis within the context of the specific theme. This method ensures the enhanced applicability and precision of the pre-trained models in sentiment analysis for texts related to specific subjects, showcasing the flexibility and adaptability of the proposed models in handling various themes and topics.

6. Conclusions

Within the scope of this study, we harnessed the capabilities of natural language processing (NLP) to dissect tweets referencing ChatGPT. Our methodology spanned diverse areas including the study of temporal patterns in tweets, extraction of content-centric attributes, topic classification/modeling, and tweets sentiment analysis. Using the Wolfram Mathematica computational platform as our foundation, we derived three primary insights. First, we delved deeply into both the temporal trends and the inherent content nuances of tweets associated with ChatGPT. Second, our research led to the development and subsequent validation of a topic classification/modeling framework tailored specifically for ChatGPT-focused tweets. Finally, we explored and evaluated four distinct sentiment analysis methodologies for tweets, all underpinned by transfer learning principles. This endeavor enriches our grasp of how ChatGPT resonates with its users and reaffirms the enduring value of such analytical pursuits in the dynamic field of NLP. Moreover, this work serves as a pivotal reference for scholars who are accustomed to using Wolfram Mathematica in other research domains, aiding their efforts in text analytics on social media platforms.

Author Contributions: Conceptualization, data analysis, methodology, software selection, validation analysis, writing—review and editing: Y.S. and Z.J.K.; coding, writing—original draft: Y.S.; supervision: Z.J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are “500 k ChatGPT-related Tweets Jan–Mar 2023” and “the sentiment140 dataset”, which were sourced from the Kaggle website. <https://www.kaggle.com/datasets/khalidryder777/500k-chatgpt-tweets-jan-mar-2023>.

Acknowledgments: We gratefully acknowledge two anonymous reviewers for their careful reading of our initial manuscript and for offering constructive and insightful critique that allowed us to significantly improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aljanabi, M. ChatGPT: Future Directions and Open possibilities. *Mesopotamian J. Cybersecur.* **2023**, *2023*, 16–17. [[CrossRef](#)]
2. Dida, H.A.; Chakravarthy, D.S.K.; Rabbi, F. ChatGPT and Big Data: Enhancing Text-to-Speech Conversion. *Mesopotamian J. Big Data* **2023**, *2023*, 31–35. [[CrossRef](#)]
3. Bian, J.; Yoshigoe, K.; Hicks, A.; Yuan, J.; He, Z.; Xie, M.; Guo, Y.; Prospero, M.; Salloum, R.; Modave, F. Mining twitter to assess the public perception of the “internet of things”. *PLoS ONE* **2016**, *11*, e0158450. [[CrossRef](#)]

4. Guo, J.; Radloff, C.L.; Wawrzynski, S.E.; Cloyes, K.G. Mining twitter to explore the emergence of COVID-19 symptoms. *Public Health Nurs.* **2020**, *37*, 934–940. [[CrossRef](#)] [[PubMed](#)]
5. Bian, J.; Topaloglu, U.; Yu, F. Towards large-scale twitter mining for drug-related adverse events. In Proceedings of the 2012 International Workshop on Smart Health and Wellbeing, Maui, HI, USA, 29 October 2012; pp. 25–32.
6. Zucco, C.; Calabrese, B.; Agapito, G.; Guzzi, P.; Cannataro, M. Sentiment analysis for mining texts and social networks data: Methods and tools. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2010**, *10*, e1333. [[CrossRef](#)]
7. Rambocas, M.; Pacheco, B. Online sentiment analysis in marketing research: A review. *J. Res. Interact. Marketing* **2018**, *12*, 146–163. [[CrossRef](#)]
8. Haque, M.U.; Dharmadasa, I.; Sworna, Z.T.; Rajapakse, R.N.; Ahmad, H. I think this is the most disruptive technology: Exploring sentiments of chatgpt early adopters using twitter data. *arXiv* **2022**, arXiv:2212.05856.
9. Abdullah, M.; Madain, A.; Jararweh, Y. Chatgpt: Fundamentals, applications and social impacts. In Proceedings of the 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS), IEEE, Milan, Italy, 29 November 2022–1 December 2022; pp. 1–8.
10. Dwivedi, Y.K.; Kshetri, N.; Hughes, L.; Slade, E.L.; Jeyaraj, A.; Kar, A.K.; Baabdullah, A.M.; Koohang, A.; Raghavan, V.; Ahuja, M.; et al. “So what if chatgpt wrote it?” multidisciplinary perspectives on opportunities, challenges and implications of generative conversational ai for research, practice and policy. *Int. J. Inf. Manag.* **2023**, *71*, 102642. [[CrossRef](#)]
11. Taecharungroj, V. “What can chatgpt do?” analyzing early reactions to the innovative ai chatbot on twitter. *Big Data Cogn. Comput.* **2023**, *7*, 35. [[CrossRef](#)]
12. Aljabri, M.; Chrouf, S.M.B.; Alzahrani, N.A.; Alghamdi, L.; Alfahaid, R.; Alqarawi, R.; Alhuthayfi, J.; Alduhailan, N. Sentiment Analysis of Arabic Tweets Regarding Distance Learning in Saudi Arabia during the COVID-19 Pandemic. *Sensors* **2021**, *21*, 5431. [[CrossRef](#)]
13. Mujahid, M.; Lee, E.; Rustam, F.; Washington, P.B.; Ullah, S.; Reshi, A.A.; Ashraf, I. Sentiment Analysis and Topic Modeling on Tweets about Online Education during COVID-19. *Appl. Sci.* **2021**, *11*, 8438. [[CrossRef](#)]
14. Roe, C.; Lowe, M.; Williams, B.; Miller, C. Public Perception of SARS-CoV-2 Vaccinations on Social Media: Questionnaire and Sentiment Analysis. *Int. J. Environ. Res. Public Health* **2021**, *18*, 13028. [[CrossRef](#)] [[PubMed](#)]
15. Macrohon, J.J.E.; Villavicencio, C.N.; Inbaraj, X.A.; Jeng, J.-H. A Semi-Supervised Approach to Sentiment Analysis of Tweets during the 2022 Philippine Presidential Election. *Information* **2022**, *13*, 484. [[CrossRef](#)]
16. Saif, H.; He, Y.; Alani, H. Semantic sentiment analysis of twitter. In Proceedings of the Semantic Web-ISWC, Boston, MA, USA, 11–15 November 2012; pp. 508–524.
17. Kiritchenko, S.; Zhu, X.; Mohammad, S.M. Sentiment Analysis of Short Informal Texts. *J. Artif. Intell. Res.* **2014**, *50*, 723–762. [[CrossRef](#)]
18. Da Silva, N.F.; Hruschka, E.R.; Hruschka, E.R., Jr. Tweet sentiment analysis with classifier ensembles. *Decis. Support Syst.* **2014**, *66*, 170–179. [[CrossRef](#)]
19. Thelwall, M.; Buckley, K.; Paltoglou, G. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.* **2012**, *63*, 163–173. [[CrossRef](#)]
20. Paltoglou, G.; Thelwall, M. Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media. *ACM Trans. Intell. Syst. Technol.* **2012**, *3*, 1–19. [[CrossRef](#)]
21. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. In Proceedings of the 52nd Annual Meeting Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; Volume 1, pp. 655–666.
22. Dos Santos, C.; Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 69–78.
23. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
24. Zhang, Y.; Roller, S.; Wallace, B.C.; Knight, K.; Nenkova, A.; Rambow, O. MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2016; pp. 1522–1527.
25. Ansari, K. Cracking the ChatGPT Code: A Deep Dive into 500,000 Tweets Using Advanced NLP Techniques. Available online: <https://medium.com/@ka2612/the-chatgpt-phenomenon-unraveling-insights-from-500-000-tweets-using-nlp-8ec0ad8ffd37> (accessed on 6 August 2023).
26. Chen, B.; Chen, X. MAUIL: Multilevel attribute embedding for semisupervised user identity linkage. *Inf. Sci.* **2022**, *593*, 527–545. [[CrossRef](#)]
27. Kim, S.W.; Gil, J.M. Research Paper Classification Systems Based on TF-IDF and LDA Schemes. *Hum. Centric Comput. Inf. Sci.* **2019**, *9*, 30. [[CrossRef](#)]
28. Shaopeng, W.; Yan, P.; Jie, W. Application Research of Text Clustering Based on LDA in Online Public Opinion Analysis. *J. Shandong Univ. Sci. Ed.* **2014**, *49*, 129–134. (In Chinese)
29. Kim, S.; Park, H.; Lee, J. Word2vec-based Latent Semantic Analysis (W2V-LSA) for Topic Modeling: A Study on Blockchain Technology Trend Analysis. *Expert Syst. Appl.* **2020**, *152*, 113401. [[CrossRef](#)]
30. Hengyi, Z.; Chenglin, L.; Tianzhu, L. A Topic Detection Method for Online Long Text. *J. Eng. Sci.* **2019**, *41*, 1208–1214. (In Chinese)

31. Shaohua, L.; Weijiang, L.; Zhengtao, Y. Research on Weibo Topic Detection Based on GV-LDA. *Softw. Guide* **2018**, *17*, 131–135. (In Chinese)
32. Available online: <https://alt.qcri.org/semEval2016/task6/> (accessed on 21 August 2023).
33. Go, A.; Bhayani, R.; Huang, L. *Twitter Sentiment Classification Using Distant Supervision*; CS224N Project Report; Stanford University: Stanford, CA, USA, 2009.
34. Pennebaker, J.W.; Francis, M.E.; Booth, R.J. Linguistic inquiry and word count: LIWC 2001. *Mahway Lawrence Erlbaum Assoc.* **2001**, *71*, 2001.
35. Padmaja, S.; Fatima, S.S.; Bandu, S. Evaluating sentiment analysis methods and identifying scope of negation in newspaper articles. *Int. J. Adv. Res. Artif. Intell.* **2014**, *3*, 1–58. [[CrossRef](#)]
36. Alessia, D.; Ferri, F.; Grifoni, P.; Guzzo, T. Approaches, tools, and applications for sentiment analysis implementation. *Int. J. Comput. Appl.* **2015**, *125*, 26–33.
37. Bonta, V.; Janardhan, N.K.N. A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. *Asian J. Comput. Sci. Technol.* **2019**, *8*, 1–6. [[CrossRef](#)]
38. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 6 July 2002; pp. 79–86.
39. Rao, G.; Huang, W.; Feng, Z.; Cong, Q. LSTM with sentence representations for document-level sentiment classification. *Neurocomputing* **2018**, *308*, 49–57. [[CrossRef](#)]
40. Le, G.M.; Radcliffe, K.; Lyles, C.; Lyson, H.C.; Wallace, B.; Sawaya, G.; Pasick, R.; Centola, D.; Sarkar, U. Perceptions of cervical cancer prevention on Twitter uncovered by different sampling strategies. *PLoS ONE* **2019**, *14*, e0211931. [[CrossRef](#)]
41. Heaton, J.; Goodfellow, I.; Bengio, Y.; Courville, A. Deep learning. *Genet. Program. Evolvable Mach.* **2017**, *19*, 305–307. [[CrossRef](#)]
42. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1817. [[CrossRef](#)]
43. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
44. Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2018; pp. 328–339.
45. Prottasha, N.J.; Sami, A.A.; Kowsher, M.; Murad, S.A.; Bairagi, A.K.; Masud, M.; Baz, M. Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors* **2022**, *22*, 4157. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.