Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.jistm.com

DEEP LEARNING-BASED CLASSIFICATION OF TOXIC ONLINE COMMENTS USING LONG SHORT-TERM MEMORY (LSTM) FOR SENTIMENT ANALYSIS

Nur 'Aqilah Aminuddin[1], Taniza Tajuddin[2*], Sofianita Mutalib[1], Siti Rafidah Muhamat Dawam[2], Noor Rasidah Ali[2], Mazura Mat Din[2]

[1] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia
Email: aqilah@uitm.edu.my, sofi@fskm.uitm.edu.my
[2] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Kedah Branch, Malaysia
Email: taniza@uitm.edu.my, srafidah192@uitm.edu.my, norrash@uitm.edu.my, mazuramd@uitm.edu.my
[*] Corresponding Author

Article Info:

Article history:
Received date: 15.10.2024
Revised date: 11.11.2024
Accepted date: 20.12.2024
Published date: 31.12.2024

To cite this document:

Aminuddin, N., Tajuddin, T., Mutalib, S., Dawam, S. R. M., Ali, N. R., & Mat Din, M. (2024). Deep Learning-Based Classification Of Toxic Online Comments Using Long Short-Term Memory (LSTM) For Sentiment Analysis. *Journal of Information System and Technology Management*, *9* (37), 352-368.

DOI: 10.35631/JISTM.937026

Abstract:

Toxic online comments have become a growing issue, spread hate and negativity, and create hostile environments that discourage constructive dialogue in online communities. They can lead to psychological distress for individuals, reduce user participation, and harm the reputation of platforms. Thus, the study aims to identify different types of toxic comments and determine whether they are positive, negative, or neutral. Analyzing various articles revealed key types of toxicity, such as obscenity, threats, severe toxicity, identity hate, and insults. A dataset comprising approximately 159,000 comments from an open-source website, specifically Wikipedia's talk page edits and thoroughly cleaned the dataset through pre-processing. Sentiment analysis was performed using the VADER Lexicon to understand sentiment polarities in these comments. Additionally, two deep learning approaches, LSTM and LSTM with GloVe word embeddings, were tested to compare the performance of both models. The data was split into an 80:20 ratio for training and testing, and tested different hyperparameters: batch sizes of 32, 64, and 128, and epochs set at 5, 10, and 15. The best results were achieved from LSTM with GloVe word embeddings, yielding an accuracy of 0.904, with a batch size of 64 and 5 epochs, and the highest precision recorded at 0.89. While the findings are promising, there is potential for improvement, including comparisons with other deep learning methods and alternative word understanding techniques.

Keywords:

Comment Classification, Deep Learning, Sentiment Analysis, LSTM, Toxic Comments

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

Introduction

As the popularity of interactive media increases tremendously, the noticing intolerable comment on social networking sites becomes a far-reaching and vital research area. The discussion is normally done in the form of comments, feedback, review, and other forms, which may be positive or negative. The positive text does not give any wrong impact, but the major challenging problem is the negative text, and it is termed as toxic (Parekh & Patel, 2017). As stated by Androcec (2020), toxic comments are defined as comments that exhibit rudeness, disrespect, or create an environment that discourages users from continuing in the discussion. Recent cases highlight the impact of hurtful language in online communities as well as on major corporations. Detecting toxic comments has been a great challenge for all scholars in the field of research and development. This domain has drawn a lot of interest not just because of the spread of hate but also people refraining other people from participating in online forums. This action affects all the creators or content providers to provide a relief to engage in a healthy public interaction which can be accessed by the public (Vidyullatha et al., 2021).

Toxic comment classification on online platforms is conventionally carried out either by moderators or with the help of text identification tools (Ozoh et al., 2019). Tools such as Google's Perspective API are used to determine the toxic comments and Yahoo's Anti-Abuse AI, uses the Aho-Corasick string pattern matching algorithm to identify offensive words. Unfortunately, the tools with certain constraints such as available in English language only and recognised comments based on the predefined set of data (Parekh & Patel, 2017). Previous works realized the complexity of this task due to intentional obfuscation of words used to avoid automated checks. It is difficult to track all racial and minority insults, and it can also be a sarcasm (Nobata et al., 2016). The advances in Deep Learning (DL) techniques, studies by researchers searching for DL whether can be used in comment classification tasks (Ozoh et al., 2019). The study emphasized that CNN and LSTM are among extensively DL in sentiment analysis, by Aken et al. (2021), making both suitable selection for this study. In line with the aims of the study, LSTM-based DL techniques will be applied to classify and detect the toxic online comments. It focuses on resolving the existing tools limitations in terms of language and complexity in identifying disguised or context-sensitive toxic content, including sarcasm and minority insults. The study attempts to foster healthier online interactions by improving the accuracy and expanding the scope of toxic comment detection systems.

Literature Review

The literature review covers four main areas: sentiment analysis, toxic comments, word embeddings, and deep learning

Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a field of Natural Language Processing (NLP) with the aim to discover methods in identifying the sentiments in text comments or based on online evaluations or opinions. The main goal is to assess the author's attitude and emotional tone related to a topic (Luo et al., 2013). In interpreting user sentiments, the most commonly technique used is by assessing public opinion, analyzing customer feedback and improve recommendations systems. The study done by Pang and Lee (2008) related to understanding the opinion mining through the expression of individual's thoughts and perspectives. These can be in the context of reviews, discussions, and social media interactions. As a result, the construct the framework of understanding sentiment dynamics and highlighting the significance of context and opinion expression relationships.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

Text expressions in source materials are often classified into two groups using sentiment analysis where one of them is facts which are objective statements regarding individuals, events, and their features. Meanwhile, the second one is opinions which are subjective expressions of sentiments, attitudes, emotions, evaluations, or feelings towards individuals, events, and their features (Luo et al., 2013). A vast volume of accessible data can be automatically analyzed using sentiment analysis techniques, allowing opinions that may assist certain customers and organizations to be extracted to accomplish their goals. The study by Farhadloo and Rolland (2016), explained the applications of sentiment analysis field by identify different levels of analysis (document, sentence, and aspect), address challenges such as sarcasm detection and compound sentence handling. In the study, various computational intelligence methods and real-world datasets from online reviews such as Yelp, TripAdvisor, employing machine learning algorithms for aspect identification and sentiment orientation. The outcomes indicate that sentiment analysis can effectively reveal patterns in customer opinions, aiding businesses in understanding consumer behavior and enhancing products.

Pavan and Prabhu (2018), reviewed sentiment classification methods using the datasets such as IMDB, Twitter and YELP and the findings highlighted the significance of classification algorithm, feature extraction and evaluation metrics. Ammar et al. (2021) used the datasets of Twitter retrieved from Kaggle to evaluate the VADER's accuracy in managing English punctuation. The integration of sentiment analysis with systems such as recommendation systems, question-answering systems and information extraction system able to enhance the performance of the system utilizing insights from sentiment analysis (Farhadloo & Rolland, 2016). The typical threshold values to categorize the sentiment include a compound score of more than and equal to 0.05 for positive sentiment, a compound score between -0.05 and 0.05 for neutral sentiment, and a compound score less than and equal to -0.05 for negative sentiment. The findings revealed that specific punctuation marks, particularly exclamation and question marks, have a significant impact on the sentiment polarity scores assigned to sentences. Another research by Youvan (2024), utilizes VADER for sentiment analysis, employs pandas for data preprocessing, and applies visualization techniques to analyze sentiment trends. The dataset used consists of titles from 2,000 academic papers, assumed to be evenly distributed over one year, stored in a text file with preprocessing steps including loading, cleaning, and date assignment.

The analysis shows significant variability in sentiment scores, reflecting different emotional tones. A rolling average highlights periods of positive, negative, and neutral sentiment, influenced by key events and seasonal trends. VADER's unique approach evaluates sentiment without prior training, using a pre-built lexicon and heuristic rules for efficient and accessible analysis. This makes VADER ideal for applications needing quick, interpretable results (Youvan, 2024). Nabila and Azlinah (2022) studied public sentiment on climate change using lexicon-based methods and machine learning classifiers like Logistic Regression, SVM, and Naïve Bayes. Their findings showed that classifier performance varied based on feature extraction techniques, and hybrid approaches improved sentiment analysis accuracy. Lexicon-based methods also provided valuable insights into climate change sentiment trends. The essential aspect in sentiment analysis is sentiment extraction to identify the polarity whether positive or negative opinions (Pang & Lee, 2008). In summary, the concepts of sentiment analysis is a powerful tool in classifying and interpreting opinions from enormous amounts of unstructured data. Next, categorize into sentiments such as positive, negative, or neutral. The

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026
applications also enhance the user experiences in online communications in interpreting emotions in textual data specifically in identifying toxic comments.

Toxic Comments

Social networking sites allow global discussions through comments, feedback, and reviews, which can be positive or negative. While positive comments are not a concern, negative ones, often called toxic, pose challenges (Parekh & Patel, 2017). Toxic comments, which can be threatening, obscene, insulting, or hate-based, create a hostile environment for users (Ozoh et al., 2019). These comments reduce user engagement and discourage participation on platforms (Aken et al., 2021), making it harder to maintain fair discussions. As a result, some platforms restrict or shut down comment sections entirely (Ozoh et al., 2019). Addressing these issues through advanced machine learning techniques is essential to classify and mitigate toxicity, thereby enabling healthier and more inclusive online discussions. This aligns with the article's focus on the impact of toxic comments in online environments.

Word Embeddings

Word embeddings play an important role in modern NLP systems where they represent words numerically in a high-dimensional space, capturing the semantic and grammatical connections between them (S. Li & Gong, 2021). They investigate automatic text classification methods utilizing deep learning and various word embedding techniques, including word2vec, doc2vec, tfidf, and an embedding layer. The study employs a dataset of approximately 50MB of news articles from Sohu news, provided by Sougo Lab. The findings reveal that the '2-layer GRU model with pretrained word2vec embeddings' achieved the highest accuracy in classifying the news text, underscoring the effectiveness of automatic text classification in handling large volumes of information.

In the context of semantic analysis, word embeddings, one of the NLP tasks offer significant insights into the meaning of the words for better understanding of word relationships (Y. Li & Yang, 2017). Patel & Tiwari (2019) described the word embedding as a method used to convert text into numerical data representation through language modelling. The word embedding assist in extracting text attributes and working as input data. They employ neural networks for words prediction based on the context, enabling the system the word relationships. Some techniques used are Word2Vec's CBOW and Skip-gram. The effectiveness of the embeddings assessed using the evaluation methods including the NN Language Model and Sparse Coding.

While the survey references multiple datasets used in word embedding research. The findings indicate that word embeddings effectively capture both semantic and syntactic information, reveal that different models have distinct strengths and weaknesses, and emphasize the necessity of robust evaluation methods to measure model performance. Some other word embedding algorithms like Global Vectors for Word Representation (GloVe) is likely to produce a state of performance achieved by neural networks. GloVe, developed by Stanford University, is a method for generating word embeddings using global word co-occurrence statistics without supervision. It combines global matrix factorization with local context methods to improve word vector quality. By training on the full word-word co-occurrence matrix, GloVe produces more accurate representations and excels in tasks like word analogy, word similarity, and named entity recognition, outperforming models like word2vec (Pennington et al., 2014). Research shows different word embedding models have unique strengths. While Word2Vec uses neural networks like CBOW and Skip-gram to predict word

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026
associations, GloVe relies on co-occurrence statistics for more precise representations. GloVe's efficiency and performance make it a powerful tool for advancing NLP research and applications.

Deep Learning

Deep learning is a specialized domain within machine learning, offers significant benefits in processing unstructured data over traditional methods. Mathew et al., (2021), specify the comprehensive overview of the deep learning techniques, its evolution, methods and various applications. The study explored training methodologies and development of frameworks, highlighting how the framework restructure network modelling without requiring extensive expertise in complex algorithms. Focusing on the key architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep Belief Learnings (DBNs) and Generative Adversarial Networks (GANs). Parekh and Patel (2017) applied machine learning approaches to detect hateful language on social media. The study addresses the complexity of identifying toxic comments, emphasizing the need for improving detection systems due to limitations of the existing system. As a result, the accuracy and reliability of online comment moderation.

Ozoh et al. (2019) utilized the dataset from Jigsaw and Wikipedia talk page edits work to propose the muti-headed model. The tasks focused on classifying toxic comments on social media platforms and developing a machine learning model. The methodology employed TF-IDF technique for text processing, classification with Logistic Regression (LR) and confusion matrix for evaluation. The findings significantly enhanced the toxic and non-toxic comments classification. Aken et al. (2021) applied a new multi-label dataset involving over 200.000 Wikipedia comments and 24.783 annotated Tweets to improve toxic comment classification. The study focused on analyzing false positives and negatives, addressing key challenges, and suggesting directions for future work. Various classifiers evaluated including Logistic Regression, bidirectional RNNs, and CNNs, combined with pretrained word embeddings. and assessed the performance using Precision, Recall, and F1-measure. The ensemble model outperformed individual classifiers, especially on the Wikipedia dataset. Key challenges included out-of-vocabulary words and long-range dependencies. Error analysis revealed patterns that improved performance on sparse classes. Various techniques, including machine learning, hybrid approaches, and lexicon-based methods, can address these challenges (Pavan & Prabhu, 2018).

Lecun et al. (2015) explain that deep learning uses models with multiple processing layers to learn data representations at different abstraction levels. The study offers an overview of deep learning, covering its techniques, applications, and impact on machine learning and AI. It

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026
discusses various deep learning methods, including NN, CNN, and representation learning, emphasizing the role of backpropagation and different learning architectures. It references applications in computer vision, speech recognition, and natural language processing that utilize large-scale datasets. The findings highlight that deep learning significantly enhances performance in tasks like object recognition, speech recognition, and language modeling, automating feature extraction and representation learning, which leads to distinguished advancements across various fields. The main focused on two factors which are nonlinear processing in multiple layers or stages and supervised or unsupervised learning where nonlinear processing in multiple layers is a method in which the current layer receives the output of the previous layer as input (Rocio Vargas et al., 2017). Moreover, deep learning is dominant in many areas and surpasses conventional machine learning techniques solely due to its ability to produce faster and more precise outcomes.

The article by Kumar & Garg (2018). aims to review the advancements in deep learning and its influence on machine learning applications. The findings indicate that deep learning greatly improves the processing of large datasets, enhances prediction accuracy, and provides superior feature extraction compared to traditional machine learning methods. Furthermore, it leverages transformations and graph technologies together to generate multi- layer learning models. Recently developed deep learning techniques have demonstrated outstanding performance across a wide range of applications, including audio and speech processing, visual data processing, NLP, among others (Alzubaidi et al., 2021). They synthesized existing literature, discussing various frameworks and libraries, such as TensorFlow, along with benchmark datasets used in deep learning tasks and emphasized deep learning's superiority in tasks like image classification, object detection, and image super- resolution. It highlights the ongoing development of deep learning and its diverse applications.

CNN is a class of deep learning method that has grown dominant in a range of computer vision tasks and is gaining interest in a variety of areas, including radiology (Yamashita et al., 2018) It discusses the architecture of CNNs, including key components such as convolution, pooling, and fully connected layers, as well as the processes involved in feature extraction and training, specifically backpropagation and gradient descent. They compared CNNs with traditional machine learning methods used in radiomics. Various datasets, particularly medical imaging datasets, are utilized in studies for training CNNs. The findings show that CNNs are capable of effectively classifying medical images and extracting features autonomously, outperform traditional methods in specific applications like disease detection. However, due to the difficult nature of CNNs, often referred to as the "black box" problem, and the necessity for large datasets to achieve optimal training results. The main advantage of CNN, it automatically identifies significant characteristics without the need for human supervision and CNNs, like conventional neural networks, were inspired by neurons in human and animal brains (Alzubaidi et al., 2021). CNNs consist of three types of layers. The first is the convolutional layer, which forms the fundamental building block of a CNN, containing a set of trainable filters or kernels. The second is the pooling layer, where the output from the convolutional layer is passed through to gradually reduce the spatial dimensions (height and width) of the representation, while preserving its depth. Finally, the last layer is the fully connected layer where the output of the CNN's feature extractor is flattened, representing a reduced form of the original input and is then sent into one or more FC layers, which are finally followed by an output layer.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

LSTMs are a form of Recurrent Neural Network (RNN) that can learn long-term dependencies while overcoming the vanishing gradient problem and to capture long dependencies in a sequence. Hence, LSTM introduces a memory unit and a gate mechanism (Okut, 2021). The application of LSTM networks for tasks involving sequential data, focusing on cancer incidence prediction and analysis. They employed LSTM architecture and used MATLAB for data preparation and training, with a data partitioning strategy of 90% for training and 10% for testing. The study utilized SEER 2017 cancer incidence data across various age groups and using COVID-19 tracking data from public sources, the results showed LSTM networks' effectiveness in predicting cancer incidence and analyzing COVID-19 trends, emphasizing the role of data quality in predictive modeling. LSTM's learning capabilities impacted both practical and theoretical domains (Houdt et al., 2020), The findings show that LSTM significantly improves tasks like speech recognition, machine translation, and sentiment analysis, highlighting its vital role in neural network advancements. LSTM comprises recurrently connected memory blocks, enabling RNNs to generate sequences for applications like music, text, and motion capture. They learn by processing real data step-by-step, predicting the next, ideal for sequence generation. In sentiment analysis, it produces output by utilizing prior computations and making use of sequential information (Patel & Tiwari, 2019).

The sentiment analysis using RNN, assess its effectiveness and utilized tools such as the Keras Sequential Model, Anaconda (Python), and Jupyter Notebook for the analysis. The research was based on the IMDB movie reviews dataset, which included 50,000 labeled reviews and 50,000 unlabeled documents. The findings revealed that the RNN model achieved an accuracy of 87.42% in sentiment analysis on the dataset. The RNN algorithm also gives each word in the input a time stamp, which maps the input order to a fixed-sized vector (Kurniasari & Setyanto, 2020). However, The RNN model analyzes text on a word-by-word basis, which can be a time-consuming process. Recent research and articles obviously reported that deep learning techniques, especially LSTM, offer significant advancements in tackling this issue, paving the way for more effective and automated systems for sentiment classification and online content moderation. Table 1 outlined and summarized the key points from the sentiment analysis, toxic comments, word embeddings and deep learning literature review.

Table 1: Summary of Key Points of the Literature Review

Authors, Year	Objectives	Techniques / Dataset	Findings
Sentiment Analysis			
Luo et al., (2013)	Incorporate sentiment analysis into social tag recommender systems	Utilized a social tagging system model and evaluated on the Dataset: MovieLens using precision metrics.	The approach enhances user experience and satisfaction by recommending items with positive feedback based on sentiment analysis.
Pang & Lee, (2008)	Review of existing literature, sentiment classification algorithms.	provide a comprehensive overview of sentiment analysis techniques.	Established a framework for sentiment analysis, emphasizing the importance of context and methodology.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

Farhadloo & Rolland, (2016)	Explore the fundamentals of sentiment analysis and its applications.	Utilized various computational intelligence methods. Datasets: Yelp and TripAdvisor.	The study found that sentiment analysis is effective in identifying patterns in customer opinions, which enhances the understanding of consumer behavior.
Oad et al., (2021)	Evaluate the performance of the VADER sentiment analyzer with English language punctuation marks.	VADER sentiment analysis tool. Dataset: Twitter downloaded from Kaggle.	Certain punctuation marks (e.g., exclamation and question marks) significantly affect sentiment polarity scores.
Youvan, (2024)	Study the functionality and effectiveness of VADER in sentiment analysis,	Utilizes VADER for sentiment analysis, employs pandas, Dataset: titles from 2,000 academic papers,	Reveals significant variability in sentiment scores, indicating diverse emotional tones. A rolling average highlights periods of positive, negative, and neutral sentiment, influenced by notable events and potential seasonal patterns.
		Toxic Comments	
Parekh & Patel, (2017)	Survey machine learning techniques for detecting hateful language on social media and challenges in toxic comment detection.	Literature review of machine learning techniques.	Identified various machine learning methods and limitations in detecting toxic comments; emphasized the need for improved detection systems.
Ozoh et al., (2019)	Classify different types of toxic comments on social media using machine learning techniques	Term Frequency-Inverse Document Frequency (TF-IDF) technique, Dataset Comments from Jigsaw and Wikipedia's talk page edits	Development of a multi-headed model capable of detecting various types of toxicity (e.g., threats, obscenity, insults, identity-based hate) and improved classification of toxic vs. non-toxic comments.
Aken et al., (2021)	Compare a classifiers on a new public	Ensemble of classifiers (Logistic Regression,	The ensemble model outperformed individual

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

	multi-label dataset of user comments and analyze false negatives and positives toxic comment classification.	bidirectional RNN, CNN) Pretrained word embeddings- Evaluation metrics, Dataset: Wikipedia (over 200,000 user comments) and Twitter (24,783 Tweets annotated with labels: hate speech, offensive	classifiers, for Wikipedia dataset. Common challenges identified include out-of-vocabulary words and long-range dependencies
Word Embeddings			
S. Li & Gong, (2021)	Explore automatic text classification methods using deep learning and various word embedding techniques	Word Embedding Methods: word2vec, doc2vec, tfidf, embedding layer - Deep Learning: 8 models including 2-layer GRU with pretrained word2vec embeddings	2-layer GRU model with pretrained word2vec embeddings achieved the highest accuracy in classifying news text, and highlights the effectiveness of automatic text classification in managing large volumes of text information
Pennington et al., (2014)	Develop a new model for word representation that combines the global matrix factorization and local context window methods	GloVe model; utilizes a word-word co-occurrence matrix. Various corpora for training, including large text datasets for evaluating word similarity and analogy tasks	GloVe achieves 75% accuracy on word analogy tasks, outperforms other models in word similarity and named entity recognition, captures semantic and syntactic regularities with faster convergence compared to models like word2vec.
Deep Learning			
Mathew et al., (2021)	Provide the deep learning techniques, various architectures, methods, and applications	Discussion of deep learning architectures (e.g., CNN, RNN, DBN, GAN), training methods, optimization techniques, and frameworks.	Deep learning outperforms especially on unstructured data and enables progressive feature learning from data at multiple levels.
Alzubaidi et al., (2021)	Review deep learning concepts, architectures, challenges, and applications	The review synthesizes and discussing various frameworks and libraries.	Highlights the superiority of deep learning in tasks such as image classification, object detection, and image super-resolution.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

Okut, (2021)	Explore the LSTM networks in deep learning.	LSTM network architecture MATLAB for data preparation and model training Data partitioning for training and testing (e.g., 90% training, 10% testing) SEER 2017 cancer incidence data for different age groups	Demonstrate the effectiveness of LSTM networks in predicting cancer incidence and analyzing trends in COVID-19 data.

Methodology

The methodology of this study consists of five phases: starting with problem formulation, followed by knowledge acquisition, data collection, data preprocessing, model development, model testing and evaluation, data visualization, and ending with documentation. Figure 2 illustrates these phases.

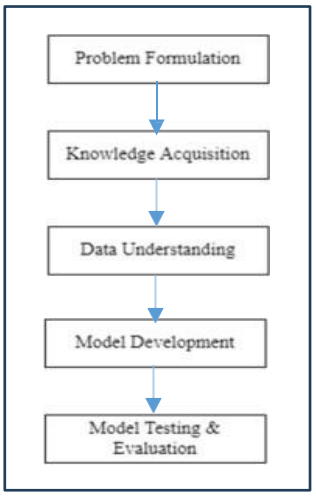

Figure 1: Phases of the Research Process

Problem Formulation

Problem formulation is the process of clearly defining the specific issue to be addressed in the research. It marks the initial stage of the study, providing a detailed explanation of the background relevant to the research domain. By the end of this phase, key elements such as the project title, background, problem statement, objectives, scope, and significance are established.

Knowledge Acquisition

Knowledge acquisition refers to the process of acquiring, structuring, and organizing information. A comprehensive literature review was conducted to gain a complete understanding of the challenges within the field and the methods employed by researchers to attain their objectives. Various sources, including websites, journals, publications, and more, were reviewed to gather insights on types of toxicity in toxic comments. All gathered

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026
information is reliable, sourced from trusted platforms such as Scopus, IEEE Xplore, and ScienceDirect. Key search terms used to identify relevant journals and articles included "Sentiment analysis," "Toxic comments," and "Deep learning". The deliverable of this phase is the literature review, which details the techniques employed in sentiment analysis, as well as an identification of the types of toxic comments.

Data Understanding

Data understanding involves identifying, gathering, and analyzing the datasets necessary to achieve the project's objectives. This phase includes key stages such as data collection and data preprocessing. The dataset which is retrieved from a website called Kaggle consist of approximately 159,000 comments. These comments were taken from various discussions on Wikipedia's talk page edits and will be processed further in the data preprocessing stage for cleaning. Various preprocessing techniques will be applied to obtain a clean dataset of toxic comments. The dataset that has been collected from Kaggle consists of 159571 data with 8 attributes which are id, comment text, toxic, severe toxic, obscene, threat, insult and identity hate. Some of the methods used for the data pre-processing are checking for missing values, removing duplicates, stopwords removal and tokenization. These activities used a few Python functions such as drop_duplicates() to remove the duplicate data, lower() to convert all characters into lowercase, stopwords and tokenizer functions. Once the data was retrieved, it was checked for missing values, and any duplicates were removed. The "Replace Missing Value" operator in RapidMiner was used for handling missing values, while the drop.duplicates() function in Python was applied to eliminate duplicates.

The collected dataset contains numerous comments from various Wikipedia talk page edits. As such, many characters present in the comments, such as special characters, non-ASCII characters, and numbers, are irrelevant to this project. According to (Mao et al., 2024), these characters do not impact sentiment analysis and removing them helps reduce noise and improve efficiency. The dataset also contained numerous instances of "\n" to indicate a new line within the comments. The re.sub() function, part of the Regular Expressions (re) module, was used to remove these instances and return a string with the desired values replaced. Next step includes stop word removal where stop word is a frequently used term such as “the,” “a,” or “an” that a search engine has been designed to ignore. Hence, these terms are removed because they occupy space in our database and use up a lot of processing time. There are also additional stopwords added such as the number words like “zero,” “one,” “two,” and so on. Tokenization is a method used in natural language processing to split texts and phrases into smaller parts that may be given meaning more easily. For instance, we can split a block of texts into either words or sentences.

Model Development

Model development is an iterative process where several models are generated, tested, and built upon until a model that meets the desired criteria is developed. Since the study focused on sentiment analysis, selecting the appropriate classification technique was essential. Numerous literature reviews from various sources have been conducted. A thorough literature review was conducted using various sources. This phase consists of two activities: sentiment extraction and LSTM model construction. Extracting sentiment was the most crucial part of this project, as the focus was on sentiment analysis. This technique was used to label the data as either positive, negative, or neutral. The approach employed for sentiment extraction was Vader (Valence Aware Dictionary and Sentiment Reasoner), which is available in the NLTK library.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026
The "compound" score was a metric that summed up all the lexicon ratings, which were normalized between -1 (most extreme negative) and +1 (most extreme positive). In this case, positive sentiment was assigned a compound score of greater than or equal to 0.05, negative sentiment had a compound score of less than or equal to -0.05, and neutral sentiment had a compound score between -0.05 and 0.05. The LSTM model was selected for this project due to its better ability to detect and capture long-term connections in data, which is crucial for understanding sentence structures. A word embedding called GloVe was also used in the model development. This was followed by the creation of the embedding matrix to analyze the frequency with which word pairs appeared together in the large corpus of text data. Finally, the pre-trained GloVe word embedding implemented into the LSTM model. By using GloVe word embeddings, the model's performance was improved, as without it, the LSTM would have needed to learn the meaning of words from scratch during training, which could lead to poor performance or overfitting.

Model Testing and Evaluation

n the model testing and evaluation phase, the model's accuracy was trained and tested. The dataset was split into two parts: a training set and a test set with an 80:20 ratio. Following the testing, the evaluation step was conducted, where the model's accuracy and loss during training were evaluated.

Results And Discussion

Data Preprocessing

Data pre-processing is the most crucial step in this project as it can bring a lot of benefits since data pre-processing eliminates missing or inconsistent data values, which can enhance a dataset's accuracy and quality, making it more reliable. Moreover, it can ensure the consistency of data. This is because it is possible to have data duplicates when using the dataset and eliminating them during preprocessing can guarantee the data values for research are consistent, which helps create accurate findings.

To remove data duplicates, the drop_duplicates() function in Python was used, and it revealed no duplicates, as the number of entries in the dataset before and after applying the function remained the same. Prior to applying this method, the dataset contained many unnecessary words, but after the method was performed, the dataset became cleaner. Although stopwords removal was performed, some unnecessary words, such as numbers, special characters, punctuation, and newline commands, remained in the dataset. These unnecessary elements were either replaced or removed. For example, "\n" and punctuation marks were replaced with white spaces, while non-ASCII characters were removed. Additionally, the dataset was converted to lowercase to facilitate easier searching.

Data Preprocessing

After the data was pre-processed, it proceeded to sentiment extraction to determine the polarity of the sentiment score. Figure 1 displays a pie chart illustrating the distribution of comments by sentiment. It reveals that out of the total number of comments, 83,662 were positive, 48,399 were negative, and 27,507 were neutral. This visual representation offers a quick and clear overview of the overall sentiment of the comments.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

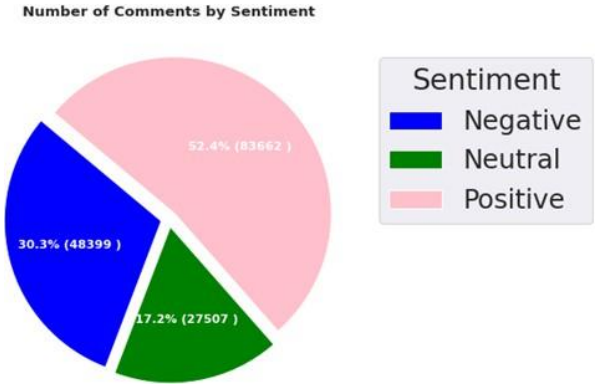

Figure 1: Number of Comments by Sentiment

A word cloud is a visual representation of the frequency of words in a set of text data. Figure 2 presents a comparison of word clouds for both negative and positive comments, visually representing the frequency of words used in each sentiment. The size of each word in the word cloud corresponds to its frequency, with more frequently used words appearing larger. This visualization offers a quick and intuitive way to identify the dominant themes and topics in the comments.

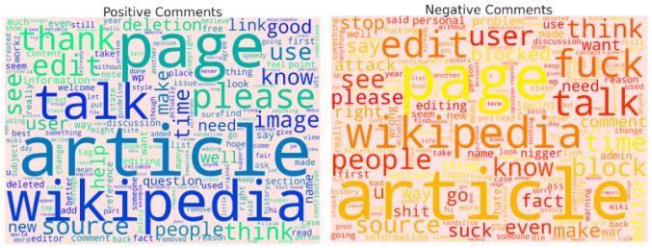

Figure 2: Word Cloud for Positive and Negative Comments

Parameter Tuning Analysis

Parameter tuning is the process of adjusting hyperparameters to improve model performance. It involves testing different values, evaluating results, and selecting the best combination. The goal is to find the most suitable hyperparameters for a specific problem and dataset. Batch size, a key hyperparameter in deep learning, refers to the number of samples processed in a single forward-backward pass through the network before updating the model. A larger batch size can speed up computation, while a smaller batch size may lead to more accurate model updates.

The analysis of Figure 3 illustrates how varying the batch size affects the performance of the LSTM model. Among the tested sizes, a batch size of 64 archives the highest overall performance as it has the highest accuracy of 0.885, precision of 0.89, recall of 0.925 and F1 score of 0.907. In contrast, a batch size of 32 demonstrates slightly lower metrics, with an accuracy of 0.858, precision of 0.831, recall of 0.904, and an F1 score of 0.865. Notably, the batch size of 128 records the lowest performance, showing an accuracy of 0.853, precision of 0.917, recall of 0.907, and an F1 score of 0.855. These findings indicate that a batch size of 64 is the best size for this LSTM model.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

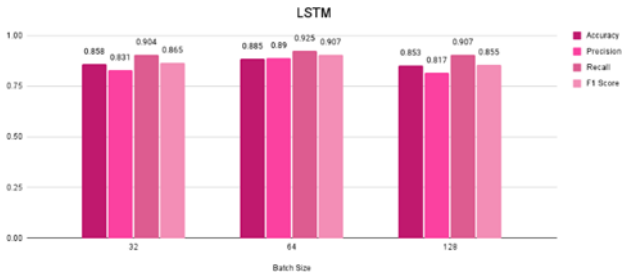

Figure 3: Analysis of Batch Size on LSTM Model

Figure 4 presents an analysis of batch sizes on the LSTM model utilizing the GloVe representation. The findings indicate that a batch size of 64 yields the highest overall performance, outperforming the other tested batch sizes of 32 and 128. A batch size of 128 has a slightly lower accuracy of 0.903 and precision of 0.876 than the batch size of 64, but a higher recall of 0.949 and F1 score of 0.911. In contrast, a batch size of 32 demonstrates the lowest performance among the three. This suggests that a batch size of 64 provides the optimal performance for this model.

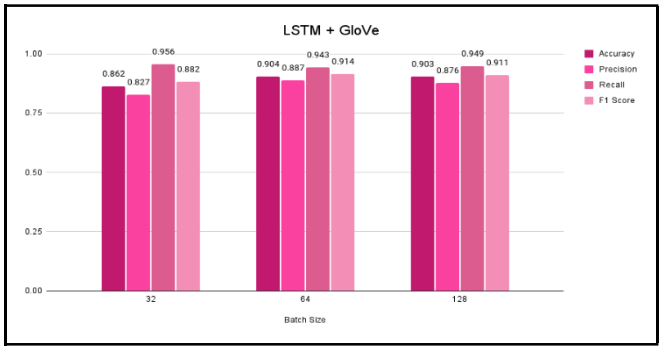

Figure 4: Analysis of Batch Size on LSTM+GloVe Model

Table 2 shows the performance of two models, LSTM and LSTM+GloVe and describes the batch size affects across four metrics: accuracy, precision, recall, and F1 score. At batch size 32, both models achieve similar accuracy, but LSTM+GloVe shows significant improvement in recall (0.956 vs. 0.904) and F1 score (0.882 vs. 0.865), indicating better balance between precision and recall. For batch size 64, all metrics improve for both models, with LSTM+GloVe achieving higher accuracy (0.904), recall (0.943), and F1 score (0.914). At batch size 128, LSTM+GloVe consistently outperforms LSTM, with improvements in accuracy (0.903 vs. 0.853) and recall (0.949 vs. 0.907), indicating its ability to handle larger datasets effectively.

Table 2: Performance Metrics for LSTM and LSTM+GloVe by Batch Size

Batch Size	Performance Metrics	LSTM	LSTM+GloVe
32	Accuracy	0.858	0.862
	Precision	0.831	0.827
	Recall	0.904	0.956
	F1 Score	0.865	0.882
64	Accuracy	0.885	0.904
	Precision	0.890	0.887
	Recall	0.925	0.943

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

	F1 Score	0.907	0.914
128	Accuracy	0.853	0.903
	Precision	0.817	0.876
	Recall	0.907	0.949
	F1 Score	0.855	0.911

Focusing on specific metrics, overall LSTM+GloVe delivers better results. Accuracy increases with batch size, with obvious improvement at batch size 128. Precision remains constant across batch sizes, but recall improves significantly with GloVe, especially at batch size 32 (0.956 vs. 0.904). These higher recall values indicate consistently better F1 scores for LSTM+GloVe, showing its ability to balance precision and recall effectively. Overall, the inclusion of GloVe embeddings enhances the performance of the model, making it a reliable choice for text classification tasks, especially where high recall and balanced metrics are important.

Figure 5 shows the analysis of the number of epochs on the performance metrics of the LSTM model. The results show that the number of epochs has a moderate effect on the model's performance. Therefore, the highest accuracy of 0.887 is achieved with 10 epochs, while the highest precision of 0.89 is achieved with 5 epochs. Moreover, the highest precision of 0.89 and F1 score of 0.907 are both achieved with 5 epochs.

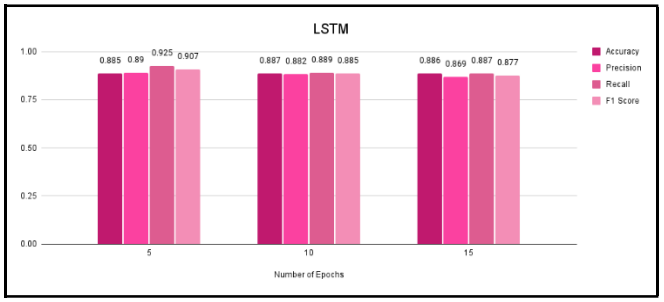

Figure 5: Analysis of Number of Epochs on LSTM Model

After going through the analysis of both hyperparameters, a batch size of 64 produced the highest performance for both LSTM and LSTM with GloVe word embedding among other batch sizes. Meanwhile in the number of epochs, the best number of epochs for this model is around 5 to 10 epochs since increasing the number of epochs beyond 10 epochs does not bring any significant improvement in the model's performance. However, the best number of epochs may change for different datasets and models, and more research may be required to discover the appropriate value for a specific situation. The results of the analysis indicate that both batch size and number of epochs are crucial hyperparameters to consider when optimizing the LSTM and LSTM with GloVe word embedding models.

Conclusion

In conclusion, the study aims to examine the sentiments associated with toxic comments. Through comprehensive evaluations and research, various forms of toxic comments such as toxic, severely toxic, identity hate, threats, insults, and obscenities have been identified. Aside from that, this study is also able to extract the sentiments in the comments and identify whether it is positive, negative or neutral. In terms of project limitations, one of them is that only one word embedding is used, which is GloVe. This is because it is trained on a large corpus of data. During the parameter tuning tests, only the LSTM model was trained, with an analysis focused

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026

on epoch size. The LSTM model with GloVe word embeddings was not trained for this parameter due to time constraints, as the single GPU available significantly extended the time required to complete even one epoch. Only the LSTM model was used, as LSTMs are particularly well-suited for this study due to their ability to make more accurate predictions regarding the sentiment of the data. In future work, it is recommended to explore other deep learning models, such as CNN or Bidirectional Encoder Representations from Transformers (BERT), and to incorporate different word embeddings like FastText and Word2Vec with the LSTM model. Additionally, experimenting with other hyperparameters, such as learning rate and activation functions, could enable a more thorough performance comparison and potentially lead to improved results.

Acknowledgment

We sincerely appreciate Universiti Teknologi MARA (UiTM) Cawangan Kedah for the invaluable support and the resources provided. We also extend our heartfelt thanks to the anonymous reviewers for their thoughtful feedback and constructive suggestions, which significantly improved the quality of this work.

References

Aken, B. Van, Risch, J., Krestel, R., & Alexander, L. (2021). Challenges for Toxic Comment Classification : An In-Depth Error Analysis. *Proceedings Ofthe Second Workshop on Abusive Language Online (ALW2)*, 33–42.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Dujaili, A. Al, Duan, Y., Shamma, O. Al, Santamaría, J., Fadhel, M. A., Amidie, M. Al, & Farhan, L. (2021). Review of deep learning : concepts , CNN architectures , challenges , applications , future directions. *Journal of Big Data*, *8*(1), 2–74. https://doi.org/10.1186/s40537-021-00444-8

Androcec, D. (2020). Machine learning methods for toxic comment classification : a systematic review ˇ. *Acta Univ. Sapientiae Informatica*, *12*(2), 205–216. https://doi.org/10.2478/ausi-2020-0012

Farhadloo, M., & Rolland, E. (2016). Fundamentals of Sentiment Analysis and Its Applications. *Sentiment Analysis and Ontology Engineering, Studies in Computational Intelligence*, *August 2018*. https://doi.org/10.1007/978-3-319-30319-2

Houdt, G. Van, Mosquera, C., & Nápoles, G. onzal. (2020). A Review on the Long Short-Term Memory Model. *Artificial Intelligence Review*, *53*(1), 1–34. https://doi.org/10.1007/s10462-020-09838-1

Kumar, V., & Garg, M. L. (2018). Deep Learning as a Frontier of Machine Learning : A Review. *International Journal of Computer Applications*, *182*(1), 22–30. https://doi.org/10.5120/ijca2018917433

Kurniasari, L., & Setyanto, A. (2020). Sentiment Analysis using Recurrent Neural Network. *International Conference on Eduaction*, 1–6. https://doi.org/10.1088/1742-6596/1471/1/012018

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *NATURE*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, S., & Gong, B. (2021). Word embedding and text classification based on deep learning methods. *MATEC Web of Conference*, *336*, 1–5.

Li, Y., & Yang, T. (2017). Word Embedding for Understanding Natural Language: A Survey. In *Studies in Big Data* (Issue 26, pp. 83–104). https://doi.org/10.1007/978-3-319-53817-4

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 9 Issue 37 (December 2024) PP. 352-368
DOI: 10.35631/JISTM.937026
Luo, T., Su, C., Xu, G., & Zhou, J. (2013). Sentiment Analysis. In *In Advances in Information Retrieval (pp. 1-17). Springer* (Issue June, pp. 53–68). https://doi.org/10.1007/978-1-4614-7202-5

Mao, Y., Liu, Q., & Zhang, Y. (2024). Journal of King Saud University - Computer and Sentiment analysis methods , applications , and challenges : A systematic literature review. *Journal of King Saud University - Computer and Information Sciences*, *36*(4), 102048. https://doi.org/10.1016/j.jksuci.2024.102048

Mathew, A., P.Amudha, & S.Sivakumari. (2021). Deep Learning Techniques : An Overview. *Nternational Conference on Advances in Computing, Communication, and Control*, 1–10. https://doi.org/10.1007/978-981-15-3383-9

Nabila, M. S., & Azlinah, M. (2022). Climate Change Sentiment Analysis Using Lexicon , Machine Learning and Hybrid Approaches. *Sustainability*, *14*(8), 1–28.

Nobata, C., Tetreault, J., Thomas, A. O., Mehdad, Y., & Yahool, Y. C. (2016). Abusive Language Detection in Online User Content. *International World Wide Web Conferences Steering Committee*, 145–153.

Oad, A., Koondhar, I. H., Butt, P. K., Oad, A., Ahmed, M., & Bhutto, S. R. (2021). VADER Sentiment Analysis without and with English Punctuation Marks. *International Journal of Advanced Trends in Computer Science and Engineering*, *10*(2), 1483–1488.

Okut, H. (2021). Deep Learning: Long-Short Term Memory. In *Artificial Neural Networks and Deep Learning - Applications and Perspective* (Issue June 2021, pp. 1–23).

Ozoh, P., Osun, O., & Adigun, A. A. (2019). Identification and Classification of Toxic Comments on Social Media using Machine Learning Techniques. *International Journal of Research and Innovation in Applied Science*, *4*(11), 142–147.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, *2*(1).

Parekh, P., & Patel, H. (2017). Toxic Comment Tools : A Case Study. *International Journal of Advanced Research in Computer Science*, *8*(5), 964-967. https://doi.org/10.26483/ijarcs.v8i5.3506

Patel, A., & Tiwari, A. K. (2019). Sentiment Analysis by using Recurrent Neural Network. *2nd International Conference on Advanced Computing and Software Engineering*, 108–111.

Pavan, K. M. R., & Prabhu, J. (2018). Role of sentiment classification in sentiment analysis : A survey. *Annals of Library and Information Studies*, *65*(September), 196–209.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe : Global Vectors for Word Representation. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.

Vidyullatha, P., Padhy, S. N., Priya, J. G., Srija, K., & Satyanjani, S. (2021). Identification and Classification of Toxic Comment Using Machine Learning Methods. *Turkish Journal of Computer and Mathematics Education*, *12*(9), 70–74.

Yamashita, R., Nishio, M., Kinh, R., Do, G., & Togashi, K. (2018). Convolutional neural networks : an overview and application in radiology. *Insights into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9

Youvan, D. C. (2024). *Understanding Sentiment Analysis with VADER: A Comprehensive Overview and Application* (Issue June). https://doi.org/10.13140/RG.2.2.33567.98726