**CCT College Dublin**

|  |  |
| --- | --- |
| **Module Title:** | *Advanced Data Analysis*  *Big Data Storage Processing* |
| **Assessment Title:** | *Integrated CA2 Semester 2 MSc in Data Analytics* |
| **Lecturer Name:** | *David McQuaid*  *Muhammad Iqbal* |
| **Student Full Name:** | *Mahmut Zahid Malkoc* |
| **Student Number:** | *2023034* |
| **Assessment Due Date:** | *17/11/2023* |
| **Date of Submission:** | *17/11/2023* |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

**Time Series Forecasting and Sentiment Analysis via Tweets**

Mahmut Zahid Malkoc   
*CCT College Dublin*Dublin, Ireland  
2023034@student.cct.ie

https://github.com/mzmalkoc2023034/S2CA2\_2023034

***Abstract—*** ***This study aims to conduct research using Twitter data with keywords such as Big Data Storage & Processing, Text Pre-Processing, Sentimental Analysis, Time Series Forecasting, and Deep Learning. The investigation will begin by examining storage and processing strategies on large datasets obtained from Twitter. Subsequently, text pre-processing techniques will be employed to clean and organize the data effectively. The goal is to extract emotional trends from user expressions through sentimental analysis methods. Time series forecasting will be utilized to understand the temporal changes in Twitter data, and deep learning techniques will be integrated to predict future trends based on these changes. This multi-method approach seeks to provide a holistic framework for understanding and analyzing the complexity and dynamics of Twitter data.***

***Keywords— Big Data Storage&Processing,*** ***Text Pre-Processıng, Sentimental Analysis, Time Series Forecasting, Deep Learning***

1. INTRODUCTION

Twitter sentiment analysis is a burgeoning field that involves the extraction of insights and opinions from tweets to gauge the overall sentiment of users. Various techniques and methodologies have been delved into by researchers to enhance the accuracy and efficiency of sentiment analysis on Twitter data.

Recent studies (Bhuta & Doshi, 2014; Giachanou & Crestani, 2017; Kharde & Sonawane, 2016; Martínez-Cámara et al., 2014; Neethu & Rajasree, 2013; Ramadhani & Goo, 2017; Saif et al., 2012; Sarlan et al., 2014; Xiaolin & Jianqiang, 2017; Zimbra et al., 2018) have shed light on the use of machine learning, deep learning, and semantic analysis techniques for sentiment analysis in the context of Twitter. These studies highlight the challenges and advancements in the field, providing valuable insights for further research.

The methodologies employed in sentiment analysis vary, encompassing text pre-processing methods (Xiaolin & Jianqiang, 2017), deep learning methods (Ramadhani & Goo, 2017), and benchmark evaluations (Zimbra et al., 2018). The diversity in approaches reflects the complexity of analyzing sentiment in the unique context of Twitter data.

In this introduction, the aim is to synthesize the knowledge from these diverse sources to underscore the significance of sentiment analysis on Twitter. As social media continues to be a prominent platform for expressing opinions, understanding the sentiments expressed in tweets becomes crucial for various applications, including marketing, public opinion analysis, and brand management.

The integration of machine learning and deep learning techniques, as explored in the referenced literature, showcases the evolving landscape of sentiment analysis. The challenges posed by the dynamic nature of language on Twitter, including slang, abbreviations, and evolving trends, necessitate continual exploration and refinement of sentiment analysis methodologies.

In conclusion, this introduction sets the stage for a comprehensive exploration of sentiment analysis on Twitter, drawing on the rich insights provided by recent research. The synthesis of various methodologies and perspectives paves the way for a nuanced understanding of the challenges and opportunities in deciphering sentiments within the vast and dynamic realm of Twitter data.

1. TEXT PRE-PROCESSING

Text pre-processing is a crucial procedure in natural language processing, aiming to clean and organize textual data for more effective utilization, comprehensibility, and integration into machine learning models.

Textual data is inherently prone to noise, encompassing unnecessary or unwanted information. The primary objective of text pre-processing is to mitigate such noise, allowing essential information to stand out.

Given that textual data often originates from diverse sources in different formats, it becomes imperative to amalgamate or normalize the text. This is crucial for ensuring consistency within datasets, contributing to more coherent and reliable outcomes.

Stop words, ubiquitous in many languages and generally devoid of meaningful content, are frequently overlooked in text mining and analysis applications. The removal of such words aids in obtaining a more focused and semantically meaningful text.

Discrepancies in case sensitivity, such as variations in uppercase and lowercase letters, or the presence of special characters, can adversely impact the accuracy of models. Transforming the text to lowercase and cleaning special characters is instrumental in mitigating sensitivity-related issues.

Tokenization, the process of segmenting text into words or abbreviations, plays a pivotal role in breaking down textual data into smaller units. This is particularly significant in extracting features for applications like text mining or natural language processing. As can be seen in figure 1 below, the most repeated words after the tokenization process are plotted as a plot.

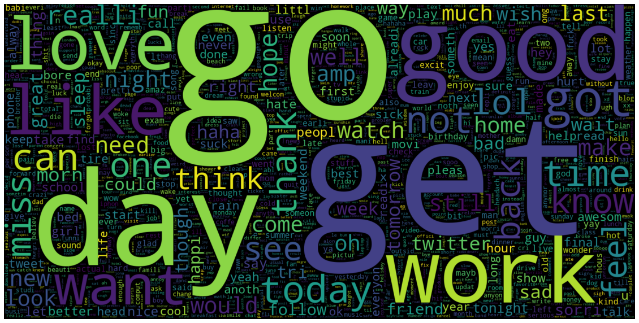


Figure 1: Most Repeating Words

The presence of special characters and numbers is often superfluous in many text-based applications. Eliminating such elements contributes to a cleaner and more focused text.

URLs or social media tags in textual data are typically extraneous for analysis purposes. Removing such information aids in obtaining a cleaner and more meaningful text.

Contraction expansion, also known as expanding contractions, denotes the procedure of transforming contracted expressions into their complete, unabbreviated counterparts. Contractions manifest as shortened versions of words or phrases, where specific letters are excluded and substituted with an apostrophe. For instance, "can't" is a contraction representing "cannot," while "I've" stands for "I have." The expansion of contractions is a prevalent practice in the realm of natural language processing (NLP) tasks, employed to uphold uniformity and establish a standardized portrayal of words within textual data.

Lemmatization serves as a linguistic process characterized by the reduction of words to their foundational or root form, referred to as the lemma. The lemma constitutes the dictionary or base manifestation of a word, and lemmatization endeavors to standardize words by eliminating inflections or variations. To illustrate, the lemma of the term "running" is "run," and the lemma of "better" is "good." Lemmatization proves advantageous in endeavors such as natural language processing, where the simplification of words to their core forms aids in feature extraction, thereby enhancing the overall efficiency of text analysis.

Stemming emerges as an alternative text normalization technique, involving the removal of suffixes or prefixes from words to derive their root or base form, recognized as the stem. Unlike lemmatization, stemming does not consistently yield valid words or lemmas; nonetheless, its primary focus lies in reducing words to a shared base that encapsulates their fundamental meaning. For example, the stem of "running" is "run," and the stem of "happiness" is "happi." Stemming often incurs lower computational costs than lemmatization, rendering it applicable in specific contexts where speed holds paramount importance, notwithstanding potential sacrifices in precision during word normalization.

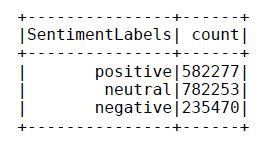


Figure 2: Number of Positive, Negative and Neutral Texts

After the text preprocessing process is completed, the number of positive, negative and neutral texts in the entire data set is shown in Figure 2 above.

According to the text pre-processing section, all the methods mentioned have been tested and implemented in the project. The most important thing in the text pre-processing section is to pay attention to the order in which the text is processed. To give a basic example, if special characters or punctuation marks are removed before URLs and hashtags, it becomes difficult to clean URLs and hashtags. In other words, URLs and hashtags turn into textual expressions and form meaningless words. For this reason, the preprocessing order is significant.

1. SENTIMENT ANALYSIS

In contemporary times, understanding the emotional tones embedded within texts is a crucial necessity for assessing human emotional responses and societal perceptions. In this context, emotional analysis encompasses a set of computational methods employed to extract and comprehend emotional expressions within texts. This article will delve into the fundamental concepts and application domains of emotional analysis.

Emotional analysis denotes the process of determining emotional content in text-based data. This process involves discerning whether words or sentences in written texts carry positive, negative, or neutral emotional content. Generally, sentiment analysis is performed using text mining and natural language processing techniques.

Key methodologies in emotional analysis include word-based analysis, machine learning models, and rule-based systems. Word-based analysis relies on matching words in the text with a pre-established emotion lexicon to determine emotion scores. Machine learning models, on the other hand, are trained on extensive datasets to acquire the ability to discern emotional expressions in texts.

Common applications of emotional analysis include social media analysis. In this domain, sentiment analysis can assist organizations in better understanding public emotional reactions, consumer satisfaction, and trends.

Some challenges in emotional analysis include polysemy, wordplay, and context dependency. Future advancements, particularly the integration of deep learning and emotional analysis techniques, hold the potential to overcome these challenges.

Emotional analysis is a critical tool for comprehending and evaluating the emotional content within text-based data. Advancements in this field will further sophisticate emotional analysis methods, enabling more effective utilization across various application domains.

1. TextBlob Method

TextBlob serves as a versatile Python tool employed for text comprehension and analysis. It functions as a utility capable of performing diverse tasks on textual data, ranging from discerning the overarching mood of a piece of writing to extracting salient information.

TextBlob boasts several noteworthy capabilities. Primarily, it possesses the capacity to assess whether a given text exhibits a positive, negative, or neutral sentiment. Functioning akin to an emotion detective, TextBlob discerns the emotional nuances embedded in words. Moreover, TextBlob aids in categorizing text into distinct groups, such as determining whether the content resembles a news article or a blog post. Another notable feature is its proficiency in extracting specific words and details from the text, encompassing names, actions, or descriptive terms a process often referred to as "feature extraction."

While TextBlob proves invaluable for straightforward tasks in gauging emotional tones in text, more intricate linguistic challenges may necessitate the utilization of sophisticated tools.

1. VADER Method

On a parallel note, "VADER" (Valence Aware Dictionary and Sentiment Reasoner) stands as another tool employed for sentiment analysis or emotion detection. The VADER Lexicon functions as a specialized dictionary within language models, containing pre-assigned sentiment scores for words. These scores, typically ranging from -1 to 1, signify the likelihood of a word or phrase possessing a positive, negative, or neutral sentiment.

During the analysis of text, VADER scrutinizes each word or phrase and assigns a sentiment score, thereby indicating whether the term is linked to a positive, negative, or neutral emotion. Notably, VADER excels in determining sentiment intensity, evaluating the strength of emotional expressions in a given text to ascertain its overall sentiment. For instance, a phrase like "very happy" might yield a higher sentiment intensity.

VADER's design is tailored for comprehending social media texts, where users frequently convey brief and swift messages. Tools like VADER prove advantageous for promptly analyzing the sentimental content inherent in such texts.

Despite VADER's proficiency in delivering rapid and straightforward sentiment analysis, its reliance on simplistic methods renders it susceptible to contextual nuances and wordplay. In certain scenarios, more intricate sentiment analysis models may be requisite.

1. TextBlob and VADER Comparison

TextBlob utilizes a simple API and is generally grounded in the Naive Bayes classification algorithm. This algorithm is predicated upon a learning process where specific words are associated with particular sentiments. VADER, on the other hand, is a sentiment analysis tool specifically tailored for short and informal texts, such as those found in social media. It employs a sentiment lexicon to determine the emotional valence of words and incorporates a set of specialized rules. While TextBlob can perform general sentiment analysis, it lacks customization for specific contexts and text types. Given its design focus on certain language usages, VADER may exhibit superior performance, particularly in scenarios like social media texts.

TextBlob serves as a general-purpose library and, consequently, may exhibit limitations in performance when processing extensive text corpora. In contrast, VADER is engineered to operate swiftly, especially on large text datasets. Both methods have been implemented in our project and subjected to comparison. It is evident that, within the framework of these fundamental differences, VADER has yielded more efficient results in our project. Consequently, sentiment scores obtained through the VADER method have been employed in the subsequent stages of the project.

1. TIME SERIES ANALYSIS

Time series analysis is a crucial statistical method for predicting future values using historical data. This analysis involves steps such as data preparation, checking for stationarity, and selecting appropriate forecasting models. Before diving into time series analysis, it is critical to format the dataset properly. This includes ensuring that date columns are appropriately separated, the data is sorted chronologically, and addressing any missing values. This step ensures that the dataset is in a suitable format for analysis.

The Augmented Dickey-Fuller (ADF) test is used to assess whether a time series is stationary or not. If there is a unit root, indicating non-stationarity, steps should be taken to make the time series stationary. Stationarity is crucial as it enhances the reliability and effectiveness of forecasting models. The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test is employed to check for stationarity. When used in conjunction with the ADF test, the KPSS test provides additional insights into stationarity, assessing whether the time series exhibits a flat trend or explosive behavior.

Linear regression is utilized to predict linear trends over time and is preferred for understanding linear relationships between variables. Its simplicity and interpretability provide an advantage in terms of model comprehensibility.

Random Forest is employed to model complex relationships by combining multiple decision trees. By capturing non-linear relationships in the dataset, it can enhance forecasting accuracy. However, due to the model's complexity, interpretability is reduced.

Following these preliminary assessments, specialized time series models such as ARMA (Autoregressive Moving Average), ARIMA (Autoregressive Integrated Moving Average), SARIMA (Seasonal Autoregressive Integrated Moving-Average), and Simple Exponential Smoothing (SES) models are applied to conduct forecasting on the dataset.

ARMA (AutoRegressive Moving Average), ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal ARIMA), and SES (Simple Exponential Smoothing) models are common forecasting models used in time series analysis. These models possess different mathematical properties and serve different forecasting purposes.

The ARMA model expresses a time series with two fundamental components: autoregressive (AR) and moving average (MA) terms. Autoregressive terms represent a regression type where past values influence the current value, while moving average terms indicate that past errors influence the current error. ARMA models are typically employed to capture the overall structures in a time series.

Similar to the ARMA model, the ARIMA model includes an integration (I) component. Integration involves taking the first or higher-order differences of the time series, rendering it stationary. ARIMA models are utilized to create a stationary time series capable of handling trends and seasonal effects.

The SARIMA model is akin to the ARIMA model but is specifically designed to address seasonal effects. Seasonal components are added to model patterns repeated at certain intervals. SARIMA is effective, especially in datasets where seasonal effects are pronounced.

The SES model is a simple exponential correction model primarily focused on the latest observations. It is employed to correct time-dependent variability and forecast future values. SES is preferred for its simplicity and low computational cost, especially when quick and direct predictions are desired.

ARMA, ARIMA, SARIMA, and SES models are fundamental forecasting models serving different purposes in time series analysis. ARMA and ARIMA are used to capture general structures, SARIMA addresses seasonal effects, and SES is suitable for straightforward and rapid predictions.

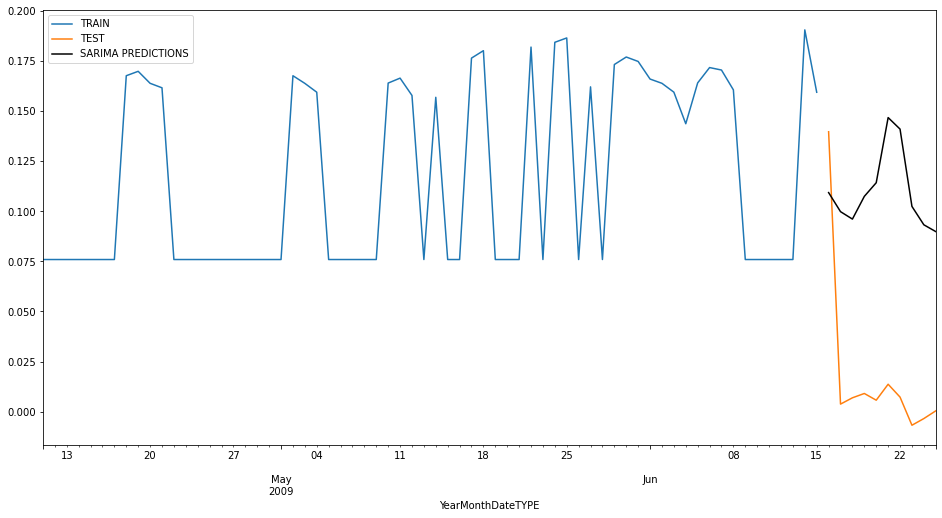


Figure 3: SARIMA Method

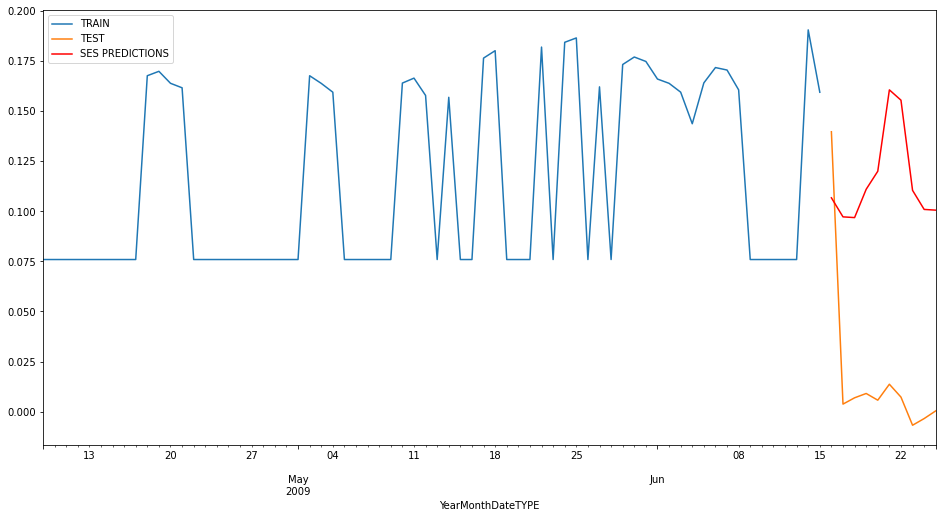


Figure 4: SES Method

Project findings, as can be seen above in Figure 3 and Figure 4, highlight the superiority of SARIMA and SES models in terms of providing accurate and reliable results. The use of these time series models has proven to be effective and helpful in achieving realistic and rational results in analysis and forecasting efforts.

1. YCSB TEST RESULTS ACCORDING TO MYSQL AND MONGODB
2. MySQL Performance Results

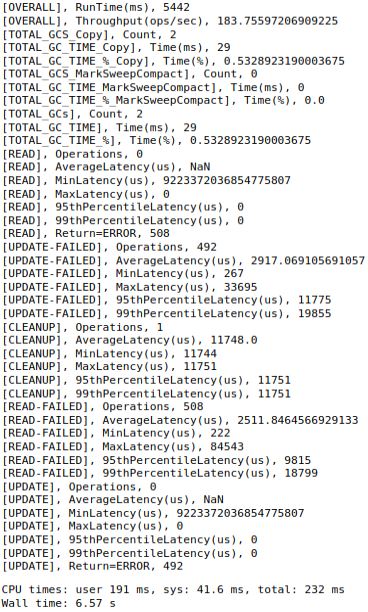
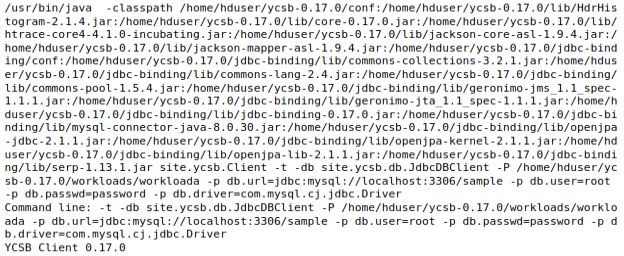


Figure 5: MySQL Performance Results

1. MongoDB Performance Results

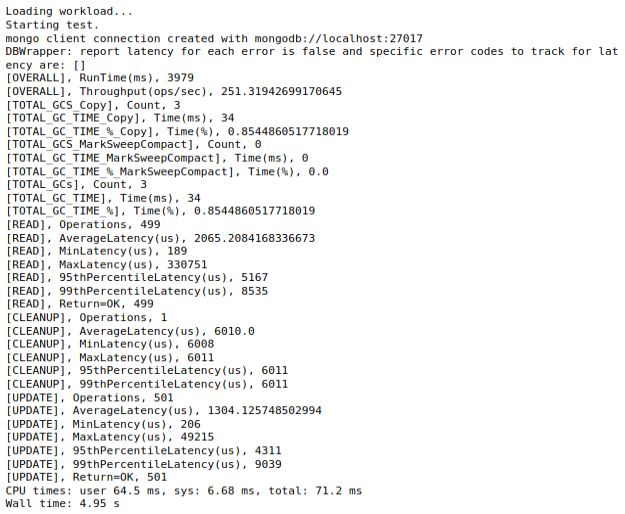


Figure 6: MongoDB Performance Results

1. MySQL and MongoDB Comparison

While Figure 5 shows YCSB Performance Results for MySQL, Figure 6 shows YCSB Performance Results for MongoDB. As can be seen Table 1 below, MySQL took 5442ms, while MongoDB took 3979ms. MongoDB has a shorter runtime, indicating faster overall performance. MongoDB has a higher throughput (251.32 ops/sec) compared to MySQL (183.76 ops/sec). Both databases have low GC impact, but MySQL has a slightly lower percentage of time spent on GC (0.53% vs. 0.85%). MongoDB performed 499 successful read operations, while MySQL had errors (508) and no successful reads. Both databases had errors in update operations (MySQL: 492 errors, MongoDB: 0 errors).

|  |  |  |
| --- | --- | --- |
| Metric | MySQL | MongoDB |
| Total Runtime (ms) | 3979 | 5442 |
| Throughput (ops/sec) | 251.32 | 183.76 |
| Total Garbage Collection Count | 3 | 2 |
| Total GC Time (ms) | 34 | 29 |
| Total GC Time Percentage (%) | 0.85 | 0.53 |
| Read Operations |  |  |
| - Operation Count | 499 | 0 |
| - Average Latency (us) | 2065.21 | NaN |
| - Min Latency (us) | 189 | 9223372036854775807 |
| - Max Latency (us) | 330751 | 0 |
| - 95th Percentile Latency (us) | 5167 | 0 |
| - 99th Percentile Latency (us) | 8535 | 0 |
| Update Operations |  |  |
| - Operation Count | 501 | 0 |
| - Average Latency (us) | 1304.13 | NaN |
| - Min Latency (us) | 206 | 9223372036854775807 |
| - Max Latency (us) | 49215 | 0 |
| - 95th Percentile Latency (us) | 4311 | 0 |
| - 99th Percentile Latency (us) | 9039 | 0 |

Table 1: MySQL and MongoDB Performance Result Comparison

In summary, MongoDB generally shows better performance in terms of throughput, read operations, and overall runtime in this particular YCSB performance test. However, the nature of the workload and specific use case requirements may influence the choice between MySQL and MongoDB.

1. CONCLUSION AND EVALUATION

The project has successfully undertaken significant steps in the realms of text processing and analysis. However, there exist areas that can be further developed, and new methodologies need to be explored.

Adapting the techniques used in text pre-processing to more specific scenarios and focusing on particular text features could enhance the results of the analysis. For instance, employing stop-word lists tailored to specific industries or subject areas could lead to a more precise text cleansing process.

The systematic comparison of sentiment analysis methods formed the foundation of the project. However, delving into the capabilities of sentiment analysis models in addressing specific language features or cultural nuances in more detail could improve overall model performance. Additionally, considering new features such as multilingual support and emotional intensity analysis could be contemplated.

The steps in time series analysis highlighted the effective use of SARIMA and SES methods. However, in future studies, the discovery of methods better adapted to different seasonal patterns or variables could enhance prediction accuracy. Furthermore, conducting a comparative study with more sophisticated methods for model selection and hyperparameter tuning could be explored.

The YCSB test results unequivocally revealed performance differences between MySQL and MongoDB. However, in future studies, tests conducted on larger datasets or different workloads could provide a more comprehensive evaluation of database performance.

In conclusion, while this project has taken a significant step, further exploration with experimental methods and analytical techniques in future studies could lead to a more profound understanding. This, in turn, could contribute to a broader spectrum of knowledge in the fields of text analysis and database performance.

REFERENCES

Bhuta, S. & Doshi, U., 02/2014. *A review of techniques for sentiment analysis Of Twitter data.* Ghaziabad, India, IEEE.

Giachanou, A. & Crestani, F., 2017-06-30. Like It or Not: A Survey of Twitter Sentiment Analysis Methods. *ACM Comput. Surv.,* 49(2), pp. 1-41.

Kharde, V. A. & Sonawane, P. S., 2016. Sentiment Analysis of Twitter Data: A Survey of Techniques.

Martínez-Cámara, E., Martín-Valdivia, M. T., Ureña-López, L. A. & Montejo-Ráez, A. R., 01/2014. Sentiment analysis in Twitter. *Natural Language Engineering,* 20(1), pp. 1-28.

Neethu, M. S. & Rajasree, R., 7/2013. *Sentiment analysis in twitter using machine learning techniques.* Tiruchengode, IEEE.

Ramadhani, A. M. & Goo, H. S., 8/2017. *Twitter sentiment analysis using deep learning methods.* Yogyakarta, Indonesia, IEEE.

Saif, H., He, Y. & Alani, H., 2012. Semantic Sentiment Analysis of Twitter. In: *The Semantic Web – ISWC 2012.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 508-524.

Sarlan, A., Nadam, C. & Basri, S., 11/2014. *Twitter sentiment analysis.* Putrajaya, Malaysia, IEEE.

Xiaolin, G. & Jianqiang, Z., 2017. Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. *IEEE Access,* Volume 5, pp. 2870-2879.

Zimbra, D., Abbasi, A., Zeng, D. & Chen, H., 2018-06-30. The State-of-the-Art in Twitter Sentiment Analysis: A Review and Benchmark Evaluation. *ACM Transactions on Management Information Systems,* 9(2), pp. 1-29.