**Southeast Europe Journal of Soft Computing** ISSN 2233 – 1859
Available online: http://scjournal.ius.edu.ba

Vol 11, No. 2, September 2022, pp. 29-39

# Kazakh Text Generation using Neural Bag-of-Words Model for Sentiment Analysis

[1*]Assel Nurlybayeva, [2]Ali Abd Almisreb, [3]Syamimi binti Mohd Norzeli, [4]Musab A. M. Ali

[1]Department of Computer Engineering and Information Security, International InformationTechnology University,
Almaty, Kazakhstan
[2]Faculty of Engineering and Natural Sciences, International University of Sarajevo,
Hrasnicka Cesta 15, Ilidža 71210 Sarajevo, Bosnia and Herzegovina
[3]Faculty of Innovative Design and Technology (FRIT), Universiti Sultan Zainal Abidin (UniSZA),
Kampus Gong Badak, 21300, Terengganu, Malaysia
[4]Faculty of Engineering, Halic University, Istanbul, Turkey
*_Corresponding Author_: nurlybayevaassel@gmail.com

Article Info

*Article history:*

Article received on 01 November 2022

Received in revised form 13 November 2022

*Keywords:*

Kazakh text generation, deep learning, sentimental analyses, Bag-of-words

**ABSTRACT:** Text generation plays an important role in making decisions in business. Analyzing the consumer's feedback provides a complete picture of the problem with a definite direction. However, sentimental analyses of reviews in the Kazakh language are not widely cultivated. In this paper, we introduce the Kazakh text generation using the Bag-of-Words model (BoW) models for analyzing the opinions of consumers in social networks. The use of proposed models in natural language processing consists of four stages: data collection, cleaning data, building model, and model evaluation. The proposed BoW model is supported by the platform - Colab notebook and implemented using the python language. Based on experimental results, defined method with higher efficiency as compared to other existing analogs.

## 1. INTRODUCTION

Natural language processing is a powerful tool for creating a clear vision for an organization [1]. Applying analyses of consumers' experiences and activities in the social network helps the economic growth of the company [2]. However, sentimental analyses for reviews containing both positive and negative opinions can lead to inaccuracy [3]. This paper focuses on solutions to this problem in the field of Kazakh language where sentimental analyses have not been widely studied yet [4].

Recently, there have been many research works conducted in the field of studying sentimental analyses in Indian, Arabic, and Turkish languages [5-7], however, there is a destitute number of studies in the Kazakh language [4]. The research declared in work [6] used machine learning methods for sentimental analyses of natural language by training models with contracting data sets according to precise features as support vector machine (SVM), Naive Bayes. Also, linguistic methods consistently use morphological analysis specifically designed sentiment dictionaries of words and phrases as well as a set of linguistic rules [5]. Additionally, morphological analysis including preprocessing methods such as tokenization, stop word elimination, stemming and POS tagging conducted in research [7] gives detailed information on the data for high accuracy in results.

To represent text for machines in work [8] used the Bag-of-words model and identified that a dynamic network is more suitable than the static network and the hybrid network for text classification. In research [9] conducted a Bag-of-words model with a support vector machine for images and determined the classifier with the best performance. Article [10] reviewed the most noteworthy work on sentiment analysis using deep learning-based architectures and presented a taxonomy of sentiment analysis. Paper [11] used the bidirectional long- and short-term memory network to determine the sentiment of a tweet and validated the performance of the proposed framework. Presence of sentimental analyses in pandemics over 10 years and the role of social media studied by Alamoodi et al. [12] to analyze the influence of technologies and science to prevent diseases like COVID-19. Garcia et al. [13] used sentimental analyses as a tool to understand public reactions to news related to COVID-19, and information dissemination in the informational field of social media. Ishihara et al. [14] investigated a score-based LR FTC system for population data synthesized with the Monte Carlo method and proved its robustness and stability of it. Authors in paper [15] proposed Machine learning models for faster delivery of justice citizens Ahuja et al. [16] used ensemble deep learning algorithms such as Naive Bayes, Random Forest, Decision Tree, and Support Vector Machine, with Bag of Words features to detect abusive comments in social media. Pandey et al. [17] in the paper discussed combinations of technics like the TF-IDF method with the BoW model to create a suitable summary for the document.

Motivated by the deficiency of works with natural language processing, the contributions of this paper are summarized as follows:

- explore and apply the BoW model to detect the sentiment of consumer feedback written in the Kazakh language: positive or negative
- analyzing BoW model with binary/ frequency/ counts/ TFIDF methods in practice.

As increasingly many people use social networks and share there their opinions, analyzing their views is important for business. The main problem is to asset beneficial tools so that businesses can consider feedback and develop. However, at present, there is a lack of approaches for considering this objective. The traditional methods of analysis take a magnificent amount of time for the company; hence, it is an ineffective way to solve the problem. To settle the problems productively this research work will be used experiences in natural language processing of other countries.

The remaining parts of the paper are organized as follows. Section II contains problem identification. Section III includes related works. Section IV describes the proposed plan. Section V is about implementation and results. Section VI contains the discussion of the implantation mentioning its advantages and shortcomings. The last Section VII concludes the paper.

## 2. PROBLEM IDENTIFICATION

One of the problems in the business of Kazakhstan which influence on profit of the company is the absence of analyses of the consumer’s feedback and consecutive decision based on that. One of the sources causing this problem is that the business uses Russian or English language to build the work in the company, by that ignore the Kazakh audience’s opinions. In this case consequences of this problem lead to negative effects for the company: a business can’t improve their performance, lose consumers and possible income. The problem can be solved in many ways for example a) changing the alphabet of the Kazakh language from Cyrillic to Latin b) manually analyzing Kazakh feedback using human sources c) using sentimental analyses of natural language processing. Injection of sentimental analyses of natural language processing can be used to save human resources and automate the analysis work of an endless stream of feedback.

## 3. RELATED WORK

This section discusses the related current work. Sentimental analysis used to analyze the opinions in the English language for UK energy company Ikoro & Victoria [18] defined that some words in different regions have the various meaning, it brings a to a solution that it is necessary to use more than on lexicons in work  As there is the same situation with the Kazakh language, these have relevance in our paper, but the problem is not lexicon, the problem is the mix of two Kazakh and Russian languages.

Anggraini et al. [19] in an Indonesian research paper for Water Company considered the importance of detecting the conjunction and its influence of it on meaning, realizing whether feedback was positive or negative [19]. The difficulties of this research come when the meaning of feedback changed several times from positive to negative and vice versa. In one more work in Indonesia Sari & Yulia [20] used Naïve Bayer’s method to analyze customer satisfaction with online transport systems and the results received 72.33% accuracy using tweets as a dataset. Because in Kazakhstan Twitter social network does not have the main part of citizens, for this research we should find another platform with people’s points of view. For Uber company Baj-Rogowska et al. [21] defined in research work that the key to success for marketing strategy and activities to raise ratings is opinion mining according to the immense knowledge hidden in Facebook's data [21]. Not many Kazakh companies understand the significance of sentimental tools for the decision-making processes in business. Abdalla & Ghazi [22] studied sentimental analyses of fast-food companies showing that the LSTM model takes less time to train and achieved high accuracy, especially on the big dataset. The difficulty is that it is hard to collect a huge amount of data in Kazakh for one company manually. In sentiment Review Analysis of the Fashion Online Industry Ernawati & Siti [23] explored the way to increase the accuracy of the Naïve Bayes algorithm with feature selection using a genetic algorithm. The limitation of this research is defining the only positive and negative types of feedback without considering neutral points of view.

## 4. SYSTEM MODEL

To evaluate reviews with sentimental analysis as negative or positive reviews developed a neural Bag-of-Words Model. The bag-of-words model is a way of performing text data in the process of modeling text with machine learning algorithms. The bag-of-words model is uncomplicated and appliance and has seen achievements in problems such as language modeling and document classification. The architecture consists of six stages and involves several utilities as depicted in Figure 1.

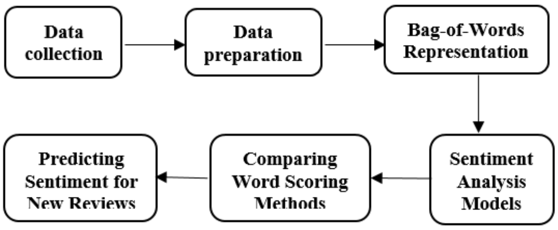
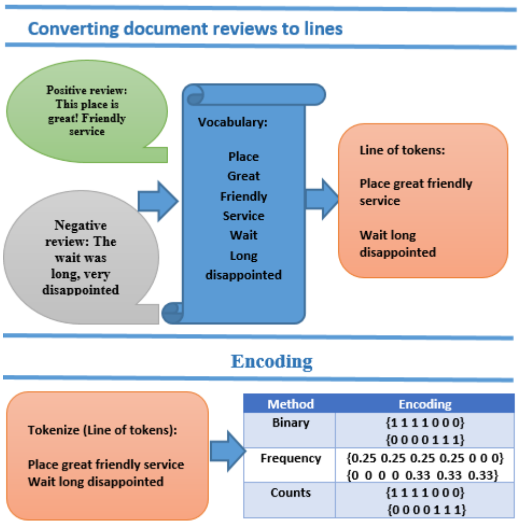

Data collection
Data preparation
Bag-of-Words Representation
Sentiment Analysis Models
Comparing Word Scoring Methods
Predicting Sentiment for New Reviews

Figure 1. Shows Architecture of Model

This section describes the process of converting each review into a representation for further rendering into the layered model. Each document is converted into a vector with a size equal to the vocabulary shape. Words in a document are calculated with binary (absence or presence), frequency, and count methods. The structure of BoW representation consists of two stages described in Figure 2:

- Modifying document reviews to lines.
- Encoding reviews to BoW vectors.

Creating a line of tokens for each document is crucial to check the existence of tokens in vocabulary. The tokenizer is covered to encode the line of tokens with the choice to select a method to score: binary (absence or presence), frequency, and counts.


Converting document reviews to lines
Positive review: This place is great! Friendly service
Negative review: The wait was long, very disappointed
Vocabulary:
Place
Great
Friendly
Service
Wait
Long
disappointed
Line of tokens:
Place great friendly service
Wait long disappointed
Encoding
Tokenize (Line of tokens):
Place great friendly service
Wait long disappointed

| Method | Encoding |
|---|---|
| Binary | {1 1 1 1 0 0 0}<br>{0 0 0 0 1 1 1} |
| Frequency | {0.25 0.25 0.25 0.25 0 0 0}<br>{0 0 0 0 0.33 0.33 0.33} |
| Counts | {1 1 1 1 0 0 0}<br>{0 0 0 0 1 1 1} |

Figure 2. Shows structure of Bag-of-Words representation

## 5. PROPOSED PLAN OF NEURAL BAG-OF-WORDS MODEL

### A. Data collection

Collected reviews are divided into positive and negative about the restaurants in Almaty city as data to test and train neural networks from the website www.2gis.com. This website the web digital maps and guides of cities and are widely spread among the Kazakh population. As 2GIS company also gives opportunity to users to comment facilities in their app possess a huge amount of feedback data to become the most preferred source to collect data.

**Algorithm 2:** Data collection process

```
Initialization: {$R[n]$: list of review data where n-number of review; $P$: positive meaning; $N$: negative meaning; $File\_p$: txt file with positive reviews; $Folder\_p$: folder with positive reviews;$File\_n$: txt file with negative review; $Folder\_n$: folder with negative reviews;
Input: { $Rn$}
Output: { $Folder\_p, Folder\_n$}
for (i=0; i<n; i++) do
If $R[i] \in P$ then
Set $File\_p$= $R[i]$
Save File_p to Folder_p
Else if
Set $File\_n$= $R[i]$
```

```
10. Save File_n to Folder_n
11. End if
12. End Elseif
13. End For
```

Algorithm 2 explained the data collection process. In step-1 initialization process of given variables is explained. In steps 2-3, input and output are shown respectively. Steps 5-7 show defining positive reviews and saving them in a text file and then into a folder with files with positive reviews. Steps 8-11 show defining negative reviews and saving them in a text file and then to the folder with files with negative reviews. Steps 6-7 and 9-10 will process in the cycle of steps 4 -13 until all reviews will be checked and saved in folders.

Set of reviews represented with equation (1):

$$D=d_1 + d_2 + d_3 + \cdots d_n \quad (1)$$

Where D-all reviews, d_1-first review, d_2-second review, d_3-third review, n- number of reviews.

Equation (1), let's mathematically reduce and represent:

$$D=\sum_{i=1}^{n} d_i \quad (2)$$

Where D-all reviews, d_i-each review, i-order of review, n- number of reviews.

Sorted to negative and positive reviews data is mathematically represented as:

$$D=\sum_{i=1}^{n/2} d\_pos_i + \sum_{i=1}^{n/2} d\_neg_i \quad (3)$$

Where D-all reviews, $d\_pos_i$-each positive review, $d\_neg_i$-each negative review, i-order of review, n-number of reviews.

Where:

$$\sum_{i=1}^{n/2} d\_pos_i = d\_pos_1 + d\_pos_2 + d\_pos_3 + \cdots + d_{pos_{n/2}} \quad (4)$$

$$\sum_{i=1}^{n/2} d\_neg_i = d\_neg_1 + d\_neg_2 + d\_neg_3 + \cdots + d\_neg_{n/2} \quad (5)$$

where, $d\_pos_i$-each positive review, $d\_neg_i$-each negative review, i-order of review, n- number of reviews.

## B. Data preparation

Text data requires appropriate preparation before the start of using it for predictive modeling. For instance, the text must be inferred to remove words, called tokenization. Also, text data must be encoded as numbers to be used as input or output for machine learning and deep learning models like the BoW model. So, data preparation concerns three phases:

- partition of data into training and test sets.

- Load and clean the data.

- Define vocabulary.

a) Separation of data into training and test sets

Aimed to create a system that can predict the mood of a restaurant review as positive or negative. Implied that data in the training set is non-identical to test data and parted as 90% to 10% respectively.

b) Load and clean the data

After loading from positive and negative files documents with one review in each document, the cleaning data process is explained for 1 document in algorithm-1 with the following stages:

- Tokenization.
- deleted punctuation.
- deleted words with no alphabetical characters.
- deleted stop words.
- deleted words with length ≤ 1 character.

**Algorithm 3:** The data cleaning process

```
1. Initialization: {$Dtrain$: review data; $T[n]$: tokens, where n-number of tokens in review
   $P$:punctuation; $A$: nor alphabetical characters; S: stop words; $Dctrain[m]$: clean review data in tokens}
2. Input: { $Dtrain$}
3. Output: { $Dctrain$}
4. T[n]=$split(Dtrain)$
5. for (i=0; i<n; i++) do
6. If $T[i] \notin P$ && $T[i] \notin A$ && $T[i] \notin S$ && $length(T[i]) >= 1$ then
7.     Set $Dctrain[m]= T[i]$
8. End if
9. End For
```

In algorithm 3, the cleaning data process is explained. In step-1 initialization process of given variables is explained. In steps 2-3, input and output are shown respectively. Step-4 split tokens on white space. Steps 5-9 show the cleaning data process, in case, if the token is punctuation, not alphabet character, in the list of stop words, or has a length less than 1 character it will not enter to cleaned train data represented in tokens. This process continues until the entire tokens of train data are checked and all the clean tokens are defined.

**Hypothesis 2:** Removing words with a length ≤ 1 character gives high accuracy in sentimental analyses.

**Proof 2:** As the words with length=1 do not have a positive or negative meaning; it will give ambiguity and accuracy will decrease.

In the tokenization process, one review consists of tokens defined in white space:

$$d_i = t_1 + t_2 + t_3 + \cdots + t_x \qquad (6)$$

Where x- number+1 white spaces in each review, $a\ t_1$- the first token of d_i-review, $a\ t_2$- the second token of d_i-review, $a\ t_3$- the third token of d_i-review,

Then equation (6) can be represented as:

$$d_i = \sum_{j=1}^{x} t_i \qquad (7)$$

Using equation (7) rewrite equation (2):

$$D = \sum_{i=1}^{n} \sum_{j=1}^{x} t_i \qquad (8)$$

Removing process to clean text:

$$CD = \sum_{i=1}^{n} \sum_{j=1}^{x} t_i \quad - \sum_{k=1}^{np} p_k - \sum_{h=1}^{nh} a_h - \sum_{w=1}^{nw} s_w \qquad (9)$$

Where $p_k$-list of punctuation, np- number of lists of punctuation, $a_h$-list of words with nor alphabetical characters, the- number of the list of words with nor alphabetical characters, $s_w$-list of stop words, nw-number of lists of stop words.

Below is represented one token $t_i$ consisting of the characters, where r-number of characters in the token.

$$t_i = c_1 + c_2 + c_3 + \cdots + c_r \qquad (10)$$

If r=1, such tokens also will be removed from the data.

c) Define vocabulary

**Algorithm 4:** Define vocabulary

```
1.  Initialization: {$Doc\_pos[n]$: n number of documents with positive review; $Doc\_neg[n]$: n number of documents with negative review; $V0$: initial vocabulary; $Vocab$: final vocabulary; $Vocab\ final$: folder}
2.  Input: { $Doc\_pos[n], Doc\_neg[n]$: }
3.  Output: { $Vocab$}
4.  for (i=0; i<n; i++)
5.      $D[m]$=Clean. data($Doc\_pos[i], Doc\_neg[i]$)
6.  End for
7.  V0[0] =D[0]
8.  for (i=0; i<m; i++)
9.      If $D[i] \notin V0$
10.     Then $V0[k]$= $D[i]$
11.     End if
12. End for
13. $Min$=2
14. for (i=0; i<k; i++)
15.     $C[i]$=V0.items()
16.     If $C[i] > Min$
17.     Then $Vocab[l] = V0[i]$
18.     End if
19. End for
20. Save $Vocab$ to $Vocab\_final$
```

In algorithm 4, the defining vocabulary process is explained. In step-1 initialization process of given variables is explained. In steps 2-3, input and output are shown respectively. Steps 4-6 show the cleaning data process with initial tokenization of entire documents in positive and negative directories according to algorithm-1. Step 7 defines the first token in primary vocabulary. Steps 8-12 check the existence of cleaned tokens in primary vocabulary and then fill primary vocabulary with unique tokens. Step 13 defines the minimum number of times one token defined in primary vocabulary can be repeated in cleaned and tokenized data in step 5. In steps, 14-19 write to final vocabulary tokens with an occurrence of more than 2. In step-20 final vocabulary list is saved into a folder for further process.

**Hypothesis 3:** Vocabulary with minimum existence of more than 2 give better performance in the programming model.

Proof 3: As all reviews are converted to vectors with the size of vocabulary, in the case of larger vocabulary the storage will be crowded, and the performance of the programming model will decrease.

The greater the vocabulary, the continued the vector depiction, hence the options for shorter vocabularies.

## C. Bag-of-Words Representation (BoW)

This section describes the process of converting each review into an illustration for further rendering into the layered model. Each document is converted into a vector with a size equal to the vocabulary format. Words in a document are accounted for with binary (absence or presence), frequency, and count methods.

**Algorithm 1:** BoW Model

```
1.  Initialization: {$Doc[m]$: m number of documents with review;$Vocab$: vocabulary;$LoT$: lines of tokens; $EL\_B[m]$: encoded lines with binary method; $EL\_F[m]$: encoded lines with frequency method; $EL\_C[m]$: encoded lines with counts method;}
```

```
2.  Input: { $Doc[m]$}
3.  Output: { $EL\_B[m]; EL\_F[m]; EL\_C[m]$}
4.  for (i=0; i<m; i++)
5.  T_pos[k] = Tokenize($Doc[i]$)
6.  if T_pos[k] ∈ $Vocab$ then
7.  Set $LoT$ = T_pos[k]
8.  Set $EL\_B[m]$ = Binary($LoT$)
9.  Set $EL\_F[m]$ = Frequency($LoT$)
10. Set $EL\_C[m]$ = Counts($LoT$)
11. end if
12. End for
```

Algorithm 1 explained the Bag-of-Wo model. In step-1 initialization process of given variables is explained. In steps 2-3, input and output are shown respectively. Step 5 used tokenize function to retrieve tokens from review. Steps 6-7 defined the line of tokens according to belonging tokens of review to vocabulary. Step 8 used the Binary function to encode lines of tokens of review to the vector. Step 9 used the Frequency function to encode the line of tokens of review to the vector. Step 10 used the Counts function to encode the line of tokens of review to the vector. Steps 4 and 12 allow encoding of each review, which means repeating steps 5-11 for all reviews.

**Hypothesis 1:** Sparse representations are harder to model.

Proof 1: It is harder to model for computational reasons (space and time complexity) and for information reasons, where the challenge is for the models to harness little information in such a large representational space. The approach is simple and flexible and can be used in a myriad of ways for extracting features from documents. A BoW is a representation of text that describes the occurrence of words within a document. It involves two things: A vocabulary of known words.

- A measure of the presence of known words.

It is called a bag-of-words because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not wherein the document. The proposed plan presented steps to make the bag-of-words model for restaurant reviews.

Definition-1: A bag-of-words is a representation of text that describes the occurrence of words within a document

- Define Vocabulary as unique variables of CD data:

$$V = \widehat{CD} = \sum_{i=1}^{n}\sum_{j=1}^{x} t_i \quad - \sum_{k=1}^{np} p_k - \sum_{h=1}^{nh} a_h - \sum_{w=1}^{nw} s_w - \sum_{r=1}^{nr} c_r \qquad (11)$$

Vocabulary consists of a list of unique tokens and equation (11) can represent:

$$V = Vt_1 + Vt_2 + Vt_3 + \cdots + Vt_f \qquad (12)$$

Then tokens in review will be represented as numbers with the binary method:

$$B = [1; 0] \qquad (13)$$

Where 1-token exists in vocabulary, 0-token doesn't exist in vocabulary. Reviews rewritten on the following conditions:

$$d_i \in V \rightarrow t_i = 1 \qquad (14)$$

$$d_i \notin V \rightarrow t_i = 0 \qquad (15)$$

It means:

$$d_i = t_1 + t_2 + t_3 + \cdots + t_x = v_i = v_1 + v_i + v_2 + \ldots v_f \qquad (16)$$

Values of review will be only numbers 1 or 0. It means the size of each review will be constant and equal to the size of the vocabulary:

$$x \neq f \qquad (17)$$

### D. Sentiment Analysis Models

Intended to develop a Multilayer Perceptron (MLP) model to predict the sentiment of encoded reviews. The model contains an input layer that equals the number of words in the vocabulary and turns the length of the input documents. To define the network using a single hidden layer and a rectified linear activation function (ReLU). The function (18) returns 0 if receives negative input, but for any positive value, x it returns that value. The model that uses ReLU is easier to train and achieves better performance.

$$f(x) = \max(0, x) \qquad (18)$$

Where x-input data.

The output layer is a single neuron with a sigmoid activation function (2) for predicting 0 for negative and 1 for positive reviews. Outputs that are much larger than 1 are transformed to the value 1, similarly, values much smaller than 0 are snapped to 0.

$$f(y) = \frac{1}{1+e^{-y}} \qquad (19)$$

Where y -output data.

The network will be trained using the efficient Adam implementation of gradient descent and the binary cross-entropy loss function (3).

To describe the Adam method provided requirements of this method as a fi with little memory requirement. The method computes individual adaptive learning rates for different parameters from estimates of the first and second moments of the gradients. Some of Adam's advantages are that the magnitudes of parameter updates are invariant to the rescaling of the gradient, its step sizes are approximately bounded by the step size hyperparameter, it does not require a stationary objective, it works with sparse gradients, and it naturally performs form of step size annealing. Initialize: $m_0$ $-1^{st}$ moment vector, $v_0$ $-2^{nd}$ moment vector, $t_0$ $-$time step:

$$\begin{cases} m_0 \leftarrow 0 \\ v_0 \leftarrow 0 \\ t_0 \leftarrow 0 \end{cases} \tag{20}$$

While $\theta_t$ not converged:

$$t \leftarrow t + 1 \tag{21}$$

Where t- time step, $\theta_t$ –resulting parameters.

Get gradients stochastic objective at time step:

$$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1}) \tag{22}$$

Where $g_t$ -gradient, $f_t$-vector of partial derivatives.

Update biased first-moment estimate:

$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) * g_t \tag{23}$$

Where $\beta_1 = 0.9$.

Update biased second raw moment estimate:

$$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) * g_t^2 \tag{24}$$

Where $\beta_2 = 0.999$

Compute bias-corrected first-moment estimate:

$$\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t\ ) \tag{25}$$

Compute bias-corrected second-moment estimate:

$$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t\ ) \tag{26}$$

Update parameters:

$$\theta_t \leftarrow \theta_{t-1} - \alpha_2 \widehat{m}_t / \left(\sqrt{\hat{v}_t} - \epsilon\right) \tag{27}$$

Where $\epsilon$ = $10^{-8}$, α-step size.

In other words, to follow the direction of the slope of the surface created by the objective function downhill until reaching a valley.

$$L = -\frac{1}{n}\sum_{i=1}^{n} y_i \ log\hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i) \tag{28}$$

Where n -output size, $\hat{y}_i$- the i-th scalar value in the model output, $y_i$ - corresponding target value.

The binary cross-entropy loss function is convenient to train a model to solve many classification problems at the same time if each classification can be reduced to a binary choice. During training and evaluation, the model keeps track of accuracy. The model on the training data fits in 10 epochs and evaluates its performance by making predictions in the test dataset and printing the accuracy.

### E. Comparing Word Scoring Methods

In this section, 4 methods for encoding words are provided by the Tokenizer function in the Keras API, such as:

- Binary
- Counts.
- Term Frequency–Inverse Document Frequency (TF-IDF).
- Freq.

In the binary method, words are represented as present (1) or absent (0) example shown in Fig.2. In the Counts method, a word is marked as an integer, where the integer is defined as the occurrence count for each word. TF-IDF method (TF — term frequency, IDF — inverse document frequency) where each word is scored based on its frequency, where words that are common across all documents are penalized. Freq method encodes words according to the frequency of the word in review.

**Theorem-1:** Binary encoding method gives more efficiency compared to frequency and counts methods in the Bag-of-words model.

**Proof-1:** According to experimental results shown in Figure 3, where displayed accuracy of methods.

**Corollary:** TF-IDF method of Bag-of-words is not stable and accurate to predict the sentiment of the review.

### F. Predicting Sentiment for New Reviews

Predicting the sentiment of new reviews involves following the same steps used to prepare the test data. Specifically, loading the text, cleaning the document, filtering tokens by the chosen vocabulary, converting the remaining tokens to a line, encoding it using the Tokenizer, and making a prediction. To predict a class value directly with the fit model by a function that returns an integer of 0 for a negative review and 1 for a positive review. To receive the predicted sentiment and an associated percentage of confidence like output.

```
Algorithm 5: Predict Sentiment for New Reviews
    1. Initialization: {$NR$: new review; R: the
       result of sentiment 1-positive review or 0-
    negative review; $P$: percentage of accuracy;
    $Vocab$: vocabulary
2.  Input: { $NR$ }
3.  Output: { $R; P$}
4.  $C = \mathbf{Clean.data}(NR)$
5.  $T = \mathbf{Tokenize}(C)$
6.  IF $T \in Vocab$
7.  $L = T$
8.  End if
9.  $B = \mathbf{TFIDF}(L)$
10. $R, P = \mathbf{Fit.Model}(B)$
```

In algorithm 5, the predicting sentiment for the new review process is explained. In step-1 initialization process of given variables is explained. In steps 2-3, input and output are shown respectively. Step 4 shows the cleaning data process for a new review. Step 5 shows the tokenization process for a new review. Steps 6-8 define a vector of new review; it means a defined line of tokens that exists in vocabulary created according to test data. In step 9 vector is encoded with the TFIDF method. In step 10 encoded vector is fitted in the model to define the sentiment of the review, if the review is predicted as positive output will be 1, otherwise, 0, also the accuracy of prediction will be given. TF-IDF method is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The tf–idf is the product of two statistics, term frequency, and inverse document frequency. There are various ways for determining the exact values of both statistics. In the case of term frequency (29), the simplest choice is to use the raw count of a term in a document, it means the number of times that term t occurs in document $d$.

$$tf(t,d) \tag{29}$$

If denote the raw count by $f_{t,d}$, then the simplest scheme is:

$$tf(t,d) = f_{t,d} \tag{30}$$

Boolean frequency if t occurs in d:

$$tf(t,d) = 1 \tag{31}$$

otherwise:

$$tf(t,d) = 0 \tag{32}$$

Term frequency adjusted for document length:

$$tf(t,d) = \frac{f_{t,d}}{number\ of\ words\ in\ d} \tag{33}$$

Logarithmically scaled frequency:

$$tf(t,d) = \log(1 + f_{t,d}) \tag{34}$$

Augmented frequency, to prevent a bias towards longer documents, raw frequency divided by the raw frequency of the most occurring term in the document:

$$tf(t,d) = 0.5 + 0.5 * \frac{f_{t,d}}{\max\{f_{t',d}: t' \in d\}} \tag{35}$$

The inverse document frequency is a measure of how much information the word provides, it means is it common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient):

$$idf(t,D) = log \frac{N}{|\{d \in D: t \in d\}|} \tag{36}$$

Where N-total number of documents in the corpus D.

Several documents where the term t appears:

$$|\{d \in D: t \in d\}| \tag{37}$$

Inverse document frequency:

$$\log \frac{\mathrm{N}}{\mathrm{n_t}} = \log \frac{\mathrm{n_t}}{\mathrm{N}} \tag{38}$$

Inverse document frequency smooth:

$$\log \left(\frac{N}{1+n_t}\right) + 1 \tag{39}$$

Inverse document frequency max:

$$\log \left(\frac{max_{\{t' \in d\}} n_{t'}}{1+n_t}\right) \tag{40}$$

Probabilistic inverse document frequency:

$$\log \left(\frac{N-n_t}{n_t}\right) \tag{41}$$

Term frequency–Inverse document frequency is calculated as:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \tag{42}$$

## 6. Experimental results

To validate the performance of scoring methods, the sentimental model is written using the Colab application based on the Python programming language. To collect data used reviews on restaurants from www.2gis.com. The laptop configuration, on which the application is executed, is described in Table 1.

Table 1. Computing environment

| Name | Description |
|---|---|
| OS | Windows 10 |
| Processor | Intel® Core™ i7-5500U CPU @ 2.40GHz × 8 |
| Processor architecture | x64 |

| | |
|---|---|
| Hard drive | 256 GiB SSD |
| RAM | 6 GiB |
| Graphics card | GeForce MX150/PCIe/SSE2 |

Based on the testing process, interesting results have been obtained as accuracy.

### A. Accuracy

As input data used 2000 documents: 1000 documents with a positive review and 1000 documents with a negative review. The vocabulary contains 25756 unique words with minimum occurrence in documents more than 2 times.

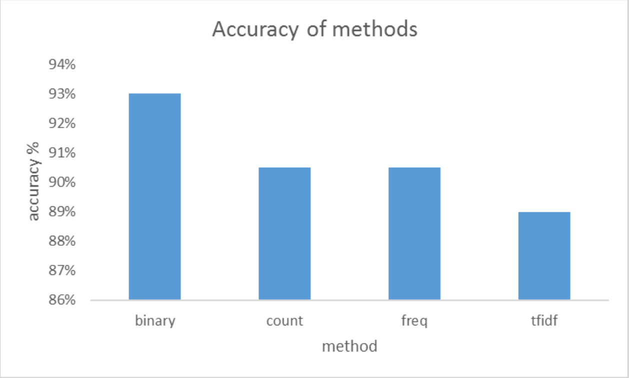


Figure 3. Shows accuracy of methods

Figures 4-7 show summary statistics for each word scoring method, by the distribution accuracy on each iteration.

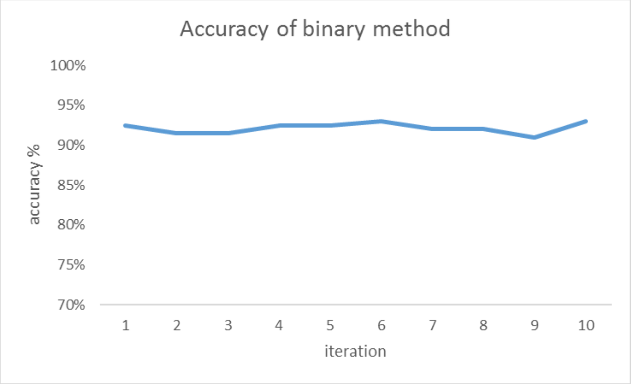


Figure 4. Shows accuracy of binary method

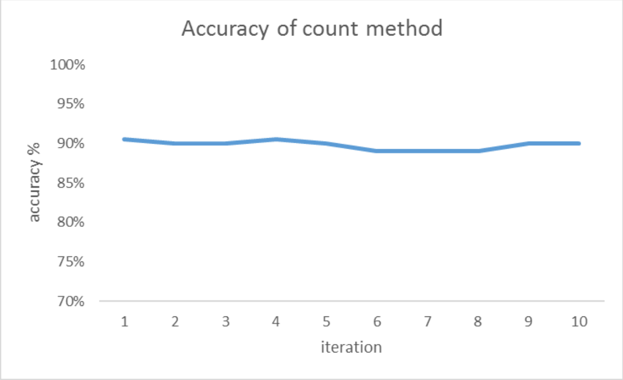


Figure 5. Shows accuracy of count method

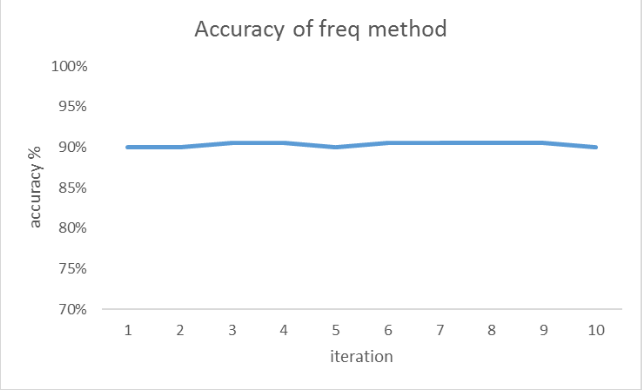


Figure 6. Shows accuracy of freq. method

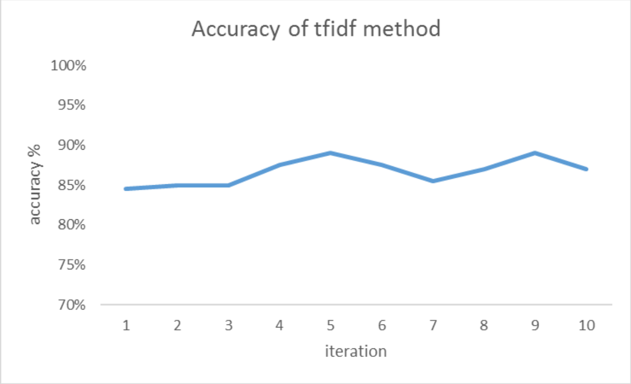


Figure 7. Shows accuracy of TF-IDF method

Made predictions for test data showed results of loss and accuracy in Figure 8 with the binary method in the sentimental model. Results of the sentimental model on test data are provided in Table 2.

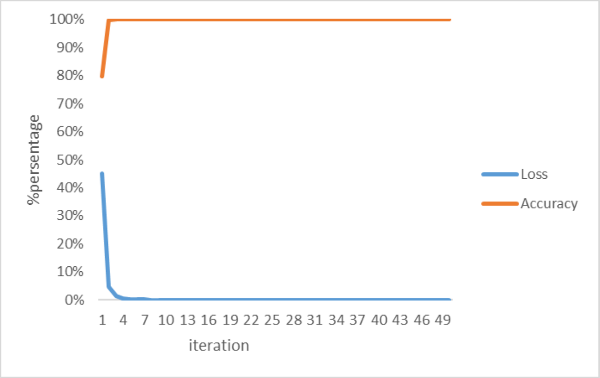


Figure 8. Shows accuracy and loss

## 7. DISCUSSION OF RESULTS

Based on the results, the performance of encoding methods on each iteration and final accuracy are demonstrated. According to Figure 3 binary method achieved the best results and become the preferred approach for the dataset. Count and freq methods have almost the same accuracy and TF-IDF is the worst. Furthermore, the accuracy of the TF-IDF method is not stable compared to the other three methods. The binary method the same time with high accuracy was also stable during all 10 iterations. Therefore, for developing a model to make predictions selected a binary method for scoring the bag-of-words model on new reviews. Figure 7 showed that the trend of loss decreases respectively to accuracy, which means the model is working correctly. According to the results in Table 2, the predicted sentiment and an associated percentage or confidence-like output are proper. The advantage of this work is that explored all required methods of the Bag of Words model and analyzed their accuracy to detect the suitable method for further research. A minor disadvantage of this work is alternate network topologies such as deeper or wider networks were not explored. As it could better performance with a more suited network. Comparing the method in this work to other latest methods as the embedding model, we can notice that synonyms in data were not defined but give almost the same accuracy.

## 8. CONCLUSION

This paper constructed the Bag of Words model to predict the sentiment of restaurant reviews as positive or negative. In process of investigation to figure out the opinion type of review cleaned data with distinct circumstances, constructed vocabulary, reviews converted to vectors, and handled the Bag-of-Words model to train a neural network for the final presuming type of test review. Conducted analysis to determine strong method with maximum accuracy among entire methods of Bag of words model. In the consequence of the study, the binary method according to accuracy revealed as the favorable method in contrast to counts, freq, and tfidf methods. Also, to minimize the amount of time to render the program planned to attempt to use the bigram to Bag-of-Words model as it scales down vocabulary and vector size and decides to affect it on the established aim or not. In conclusion, the binary method in the Bag-of-Words model has the chief accuracy to forecast the thought of reviews. In prospect will be attractive to use bigrams and identify the impact of it on the efficiency of results.

## Appendix

Table 2. Shows sentiment results of the review

| Review | Sentiment |
|---|---|
| Best restaurant!<br>Delicious | POSITIVE (56.756%) |
| Awful place | NEGATIVE (71.090%) |

## REFERENCES

[1] Yemm, G. (2006), "Can NLP help or harm your business?"

[2] Ranjan, Sandeep, Sumesh Sood, and Vikas Verma. "Twitter sentiment analysis of real-time customer experience feedback for predicting growth of Indian telecom companies." 2018 4th International Conference on Computing Sciences (ICCS). IEEE, 2018.

[3] Liu, Bing. "Sentiment analysis: A multi-faceted problem." IEEE Intelligent Systems 25.3 (2017): 76-80.

[4] Yergesh, Banu, Gulmira Bekmanova, and Altynbek Sharipbay. "Sentiment analysis on the hotel reviews in the Kazakh language." 2017 International Conference on Computer Science and Engineering (UBMK). IEEE, 2017.

[5] Phani, Shanta, Shibamouli Lahiri, and Arindam Biswas. "Sentiment analysis of tweets in three Indian languages." Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016). 2016.

[6] Baly, Ramy, Georges El-Khoury, Rawan Moukalled, Rita Aoun, Hazem Hajj, Khaled Bashir Shaban, and Wassim El-Hajj. "Comparative evaluation of sentiment analysis methods across Arabic dialects." Procedia Computer Science 117 (2017): 266-273.

[7] Yildirim, Ezgi, Fatih Samet Çetin, G. Eryigit and Tanel Temel. “The Impact of NLP on Turkish Sentiment Analysis.” (2016).

[8] D. Yan, K. Li, S. Gu and L. Yang, "Network-Based Bag-of-Words Model for Text Classification," in IEEE Access, vol. 8, pp. 82641-82652, 2020, doi: 10.1109/ACCESS.2020.2991074.

[9] Jin, Wei, and Yunsong Feng. "Automatic Classification for Ground Targets under Complex Background Based on Bag of Words Model." In IOP Conference Series: Materials Science and Engineering, vol. 711, no. 1, p. 012089. IOP Publishing, 2020.

[10] Yadav, Ashima, and Dinesh Kumar Vishwakarma. "Sentiment analysis using deep learning architectures: a review." Artificial Intelligence Review 53, no. 6 (2020): 4335-4385.

[11] Naseem, Usman, Imran Razzak, Katarzyna Musial, and Muhammad Imran. "Transformer based deep intelligent contextual embedding for Twitter sentiment analysis." Future Generation Computer Systems 113 (2020): 58-69.

[12] Alamoodi, Abdullah, Bilal Zaidan, Aws Zaidan, Osamah Albahri, Khaled Mohammed, Rami Malik, Esam Almahdi et al. "Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review." Expert systems with applications (2020): 114155.

[13] Garcia, Klaifer, and Lilian Berton. "Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA." Applied Soft Computing 101: 107057.

[14] Ishihara, Shunichi. "The Influence of Background Data Size on the Performance of a Score-Based Likelihood Ratio System: A Case of Forensic Text Comparison." ALTA 2020: 21.

[15] Saha, Dipanjan, Riya Sil, and Abhishek Roy. "A Study on Implementation of Text Analytics over Legal Domain." In Evolution in Computational Intelligence, pp. 561-571. Springer, Singapore, 2021.

[16] Ahuja, Ravinder, Alisha Banga, and S. C. Sharma. "Detecting Abusive Comments Using Ensemble Deep Learning Algorithms." In Malware Analysis Using Artificial Intelligence and Deep Learning, pp. 515-534. Springer, Cham, 2021.

[17] Pandey, Preksha, Jatin Keswani, and Subrat Kumar Dash. "Comparative Analysis of Various Techniques Used to Obtain a Suitable Summary of the Document." In Rising Threats in Expert Applications and Solutions, pp. 627-633. Springer, Singapore, 2021.

[18] Ikoro, Victoria, Maria Sharmina, Khaleel Malik, and Riza Batista-Navarro. "Analyzing sentiments expressed on Twitter by UK energy company consumers." In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 95-98. IEEE, 2018.

[19] Anggraini, Auliya, Entin Martiana Kusumaningtyas, Ali Ridho Barakbah, and M. Tafaquh Fiddin Al Islami. "Indonesian Conjunction Rule Based Sentiment Analysis For Service Complaint Regional Water Utility Company Surabaya." In 2020 International Electronics Symposium (IES), pp. 541-548. IEEE, 2020.

[20] Sari, Eka Yulia, Akrilvalerat Deainert Wierfi, and Arief Setyanto. "Sentiment Analysis of Customer Satisfaction on Transportation Network Company Using Naive Bayes Classifier." 2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM). IEEE, 2019.

[21] Baj-Rogowska, Anna. "Sentiment analysis of Facebook posts: The Uber case." 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE, 2017.

[22] Abdalla, Ghazi, and Fatih Özyurt. "Sentiment Analysis of Fast Food Companies With Deep Learning Models." The Computer Journal (2020).

[23] Ernawati, Siti, and Eka Rini Yulia. "Implementation of The Naïve Bayes Algorithm with Feature Selection using Genetic Algorithm for Sentiment Review Analysis of Fashion Online Companies." 2018 6th International Conference on Cyber and IT Service Management (CITSM). IEEE, 2018.