Scientific Journal of Informatics
Vol. 9, No. 1, May 2022
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Topic Modeling on WhatsApp User Reviews Using Latent Dirichlet Allocation

Iqbal Kharisudin[1*], Hera Masri'an[2]
[1,2]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

Abstract.
Purpose: Topic modeling is a practical algorithm for identifying topics in text data. This study aims to find issues of WhatsApp user reviews using Latent Dirichlet Allocation (LDA) and describe the characteristics of each case.
Method: We used 1710 WhatsApp user reviews written 7-13 August 2020 on Google Play. This research was conducted with a qualitative method consisting of five stages: problem identification, data retrieval, preprocessing, modeling, and analysis. The modeling stage consists of making a Document-Term Matrix (DTM), determining the number of iterations and topics, and building a model. We use perplexity as to the indicator in determining the number of iterations and topics. A lower perplexity value indicates a better model performance. The analysis phase includes observations on the top terms and documents to label and describe the characteristics of each topic.
Result: Topic modeling produces word-topic and document-topic assignments. The word-topic assignment contains words with high probability (top terms). Document-topic assignment reveals documents that have a high probability (top documents). The topics most frequently discussed were voice and video calls with 104 reviews, 86 reviews of call quality, photo and video quality with 100 reviews, and voice messages with 75 reviews.
Novelty: In this research, a topic model has been generated for a user review of the WhatsApp application using Latent Dirichlet Allocation. The number of iterations in the modeling was determined based on the observation of the perplexity value, instead of randomly assigning iterations.

Keywords: Text Mining, Topic Modeling, Latent Dirichlet Allocation, WhatsApp User Review
Received February 2022 / **Revised** February 2022 / **Accepted** April 2022

INTRODUCTION

Since the widespread use of the internet in the 1990s, human activities have generated large amounts of data [1]. The data contains varied information and the volume increases rapidly in a relatively short time. Such data phenomena are often referred to as big data [2]. The problem of how to utilize the data is the background for the birth of data mining. Han & Kamber [3] call this data mining as a process of finding knowledge from large-scale data. Mining work with text data is called text mining [4]. Text mining is defined as the process of obtaining information from an unstructured text database [5]. Text mining provides various solutions for processing, organizing, and analyzing large amounts of unstructured text [6].

Learning algorithms in text mining are divided into supervised and unsupervised learning. In supervised learning, each record in the data set has a label. Unlike supervised learning, unsupervised learning algorithms do not have labels on their data sets. One of the unsupervised learning algorithms is clustering. In a collection of documents, this clustering method allows faster retrieval of information because the document will be checked automatically [7]. One of the popular clustering methods is topic modeling or topic modeling. Topic modeling works by dividing (grouping) the entire input data into several groups based on the similarity of terms owned by each document. A researcher or analyst may not want to work with all of the data, but only take some of the data on a particular topic. By categorizing the corpus into smaller sub-corpuses based on the topic, data analysis can be carried out in a more focused and efficient manner as needed. In addition, the processed data will be lighter (less) so the algorithm can work faster.
There have been many studies on the topic of modeling covering various techniques and problems. Esmaeili et al. [8] apply topic modeling to study the topic of a review text based on certain aspect groups using Variational Aspect-based Latent Topic Allocation (VALTA). Other techniques, such as Natural Language

*Corresponding author.
Email addresses: iqbalkharisudin@mail.unnes.ac.id (Kharisudin), hera.masrian@students.unnes.ac.id (Masri'an)
DOI: 10.15294/sji.v9i1.34941

Processing (NLP) are also used to extract information in the form of a summary of the topics discussed in a collection of health product review texts [9]. Besides VALTA and NLP, the technique that is often used is Latent Dirichlet Allocation (LDA). In general, LDA can identify topics (often discussed) in text data [10]. In VALTA and NLP, the number of topics is determined based on several conditions involving the subjective opinion of humans (researchers). This can lead to inconsistent results because human intuition and understanding differ. Determination of the number of topics in the LDA is based on a value called perplexity. After this value is obtained, a researcher only needs to observe the candidate number of topics that displays the lowest perplexity value [11]. Thus, the results obtained will be consistent even though it is done by different people. In addition to these advantages, the LDA technique has also been proven to be able to analyze text from large documents [12].

LDA has been widely applied in various studies, such as by Annisa et al. [13] who use LDA to study information in a hotel visitor review. In the research results, topic modeling can identify eight topics along with the words that make up those topics. Another study using LDA was conducted by Sutherland et al. [14] by successfully describing topics about the quality of accommodation services in tourism based on customer reviews. LDA can also be applied to find out trending topics in a collection of customer complaints [15] and identify topics from a collection of online news titles [16]. In politics, LDA can be used to analyze political issues such as the presidential election through Twitter posts [17]. For the health care sector, topic modeling with LDA can provide recommendations for the best doctors to treat certain diseases based on reviews or experiences from previous patients [18]. LDA can also be applied in software engineering, such as the detection of malicious android applications based on their description [19].

Smartphone application development companies certainly want to satisfy users of their products and services. Therefore, improvements and innovations are continuously made to meet user demands. This needs to be done because the quality of a product and service affects customer/user satisfaction [20]. To find out complaints and feedback, the service provider provides the opportunity for users to write their personal reviews of the services or products that have been used. User reviews can be used as material for analysis so that useful information is obtained. One of the most popular applications is WhatsApp, which is now used by more than two billion people in 180 countries. The developer uses such a large number of users to obtain information to improve service quality by collecting user criticism and suggestions. Quoting from Google, there have been more than one hundred million user reviews and this number is growing. Such user reviews can be useful information if studied. However, the number of one hundred million is not small and of course impossible if it is checked manually. What's more, the reviews written cover some different topics. With so many reviews, it's easier to group them by topic first. Therefore, text mining is needed to identify and analyze the topic of the review more quickly.

Based on this background, the purpose of this research is to develop and find a suitable topic model for WhatsApp user reviews. This study uses Latent Dirichlet Allocation (LDA), which is one of the topic modeling techniques that utilize a probabilistic generative model for discrete data sets in the form of text, where each item in the collection is not only on one topic, but has a probability of membership in several topics [11]. Topic modeling can help identify what topics appear in a collection of texts [21], so it can be seen the things that are often mentioned or complained about from the WhatsApp application service. The selection of WhatsApp as the object of research is expected to be a material consideration for other companies in developing products that are being pioneered, considering that the application is prevalent. Before modeling, the number of iterations and topics will be determined for the model to be built. After obtaining the topics for each document, an analysis of the terms that appear in each topic will be carried out. The study was carried out to obtain information in the form of topic descriptions that could be understood more easily. The results of the analysis of topic modeling using LDA are expected to provide information about WhatsApp users in a brief review often discuss topics.

METHODS

This research was conducted using a qualitative method consisting of five stages (as shown in Figure 1), namely problem identification, data collection, preprocessing, modeling, and analysis of results. At the problem identification stage, research is carried out on the data to be used and the specifications of the appropriate research tools for data collection. The method used to obtain the data is web scraping. This method was chosen because it can handle the problem of retrieving data from online sources [22]. Web scraping requires a stable internet connection but can work faster and more efficiently than regular copy-paste [23]. The R and RStudio application were used to assist data retrieval and processing in this study.

The preprocessing stage aims to organize and prepare data to match the application library used. The data used in this study is the text of WhatsApp user reviews in English and accessed via Google Chrome. Based on the scraping results, 11,960 reviews were obtained which included username, date of review, rating or star, and review text. In order to limit the topic of the review not being too broad, 1,710 reviews were selected that were written on 7 – 13 August 2020. Names, stars, and dates of review will not be used in this study. The data obtained from the scraping process is saved in .csv format. To simplify the process of organizing, the format was changed to .xls. Emojis and emoticons are automatically converted to UTF-8 format.

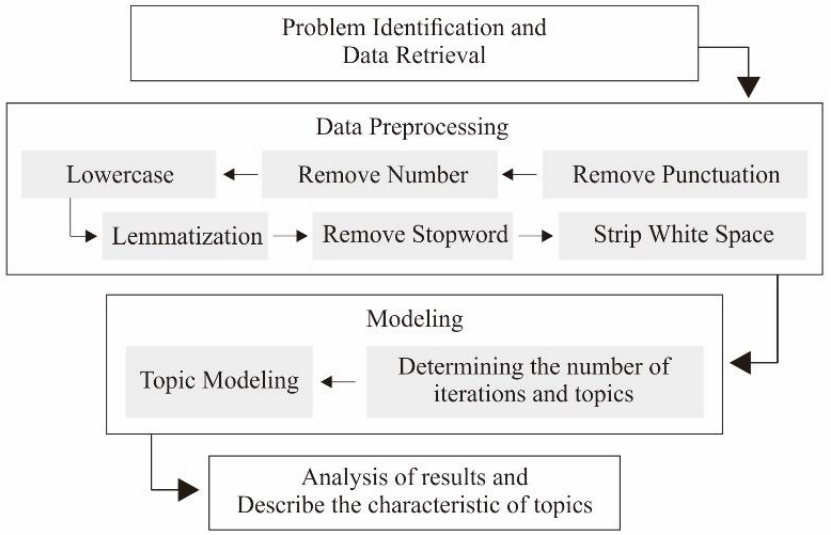

Figure 1. Research flowchart

The LDA algorithm requires input in the form of a Document-Term Matrix (DTM) with a weighting of Term Frequency (TF), which is the default weighting for clustering algorithms. Data from the previous stage will be converted as a corpus and processed into DTM using the package tm. The process of calculating the model in this study mainly uses the functions contained in the package topic models and quanteda, the rest is helped by the package tidy verse for processing graph plots, and wordcloud2 in the visualization section. Before working on the main algorithm, the number of topics and iterations must be determined and validated. An optimum model can be produced which is considered good enough to represent the existing topics [22]. The process of determining the candidate number of iterations and topics begins by dividing the DTM into two parts, namely training and testing data. Next, a model will be built using training data and applied to testing data [24]. LDA is one of the unsupervised learning which consists of a probabilistic generative model [25]. This generative process defines a joint probability distribution of random variables, both observed and hidden variables. The combined distribution aims to calculate the conditional distribution of the hidden variables obtained from the observed variables. This conditional distribution is often called the posterior distribution. The computational problem in deducing the topic structure of a document is how to calculate the posterior distribution [26].Te 2. LDA generative process

The graphic model in Figure 2 [27] is a representation of the generative process performed by the LDA algorithm. The shaded circle is an observed variable, while the unshaded circle is a hidden (latent) variable. Parameters α determine the distribution of topics in the corpus. The larger the α value, the corpus contains a wider mix of topics. The parameter β determines the distribution of words in the topic. A high β value indicates more word mixes in the word distribution (ϕ). Variable θ represent the distribution of topics in each document. The larger the θ value, the more topics that appear in the document. The z variable declares the topic for a document (collection of words). Meanwhile, the w variable represents a term. The outer box represents an iterative process for determining the proportion of document topics [12]. Parameters α and β are parameters of the corpus level with the assumption that sampling is taken during the process of compiling the corpus. Variable θ and ϕ resides at the document level with sampling in each document. Then, the variables z and w are at the term level and are sampled once for each word in each document [11]. With regard to the generative processes in the LDA, Grün & Hornik [28] describe it as follows.

1) Choosing the term distribution for each topic with $\phi \sim$ Dirichlet(β); Each topic can be defined as the probability distribution for each word in the vocabulary as many as V. Technically, for each topic $k = \{1,2, \dots, K\}$, we can define the ϕ_k value of a Dirichlet distribution with the main parameter β.
2) Choosing the proportion of topic distribution for each document, $\theta \sim$ Dirichlet(α); Each document in the corpus can be expressed as a document probability distribution for each topic (θ). This distribution is obtained from the Dirichlet distribution with the prior parameter α.
3) Furthermore, for each d document and N word of w_n,
   a) Choose a topic z_n from the topic-document distribution (θ), $z_n \sim$ Multinomial(θ).
   b) Choose a word w_n from the word distribution (ϕ_k), with z_n as k taken from the previous step, $P(w_n|z_n, \beta)$.

According to Blei, et al [11], the above generative process can be explained by several equations. If there are parameters α and β, then the joint distribution of a mixture of topic proportions θ, word proportions ϕ, N sets of z topic, and N word sets in the document d, is written in equation (1), where: $P(z_n|\theta)$ only for θ_i with a unique i such that $z_n^i = 1$; The $P(\theta|\alpha)$ value in equation (2) below is a Dirichlet distribution which gives the probability density value for a random variable θ with dimension $k = \{1,2, \dots, K\}$, parameter $\alpha_i > 0$, and $\Gamma(x)$ is a Gamma function

$$P(\theta, z, d|\alpha, \beta) = P(\theta|\alpha) \prod_{n=1}^{N} P(z_n|\theta) P(w_n|z_n, \beta) \quad (1)$$

$$P(\theta|\alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \theta_1^{\alpha_1 - 1} \dots \theta_K^{\alpha_K - 1} = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}. \quad (2)$$

By integrating equation (1) with respect to θ and adding each z, we get the marginal distribution for a single document in equation (3). Then, by taking the product of the marginal probability of a single document from equation (3) above, the probability of a corpus is obtained with equation (4), K indicates the number of topics, M is the number of documents, N represents the number of terms [11]

$$P(d|\alpha, \beta) = \int P(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} P(z_n|\theta) P(w_n|z_n, \beta) \right) d\theta \quad (3)$$

$$P(D|\alpha, \beta) = \prod_{m=1}^{M} \int P(\theta_m|\alpha) \left(\prod_{n=1}^{N_m} \sum_{z_{m,n}} P\left(z_{m,n}|\theta_m\right) P\left(w_{m,n}|z_{m,n}, \beta\right) \right) d\theta_m. \quad (4)$$

In computational problems, conclusions about the structure of the topic are carried out by calculating the posterior distribution in equation (5) [11], or equation (6) [29] below. The LDA algorithm assumes that words will be generalized based on topics (fixed conditional distribution) and these topics can be exchanged infinitely in a document. Based on de Finetti's theorem, the probabilities of the word sequences and topics are obtained through the following equation (7). de Finetti's theorem states that the combined distribution of infinitely exchangeable sequences of variables is a random parameter taken from several distributions with independent variables and identically distributed. An infinite sequence of random variables is said to be infinitely exchangeable if every finite subsequence is exchangeable. A finite set of random variables $\{z_1, z_2, \dots, z_N\}$ is exchangeable if its combined distribution does not change in its permutations. If π is the permutation is an integer from 1 to N, then $P(z_1, z_2, \dots, z_N) = P\left(z_{\pi(1)}, \dots, z_{\pi(N)}\right)$ [11].

$$P(z|d) = \frac{P(d,z)}{\sum_z P(d,z)} \quad (5)$$

$$P(\theta, z|d, \alpha, \beta) = \frac{P(\theta, z, d|\alpha, \beta)}{P(d|\alpha, \beta)} \quad (6)$$

$$P(d, z) = \int P(\theta) \left(\prod_{n=1}^{N} P(z_n|\theta) P(w_n|z_n) \right) d\theta. \quad (7)$$

To find topics in the corpus $D = (d_1, d_2, \dots, d_M)$, where each word w_v is in several document d_m, an β estimate must be obtained that can produce a high probability value for the words that appear. One way to get this estimate is to try to maximize $P(d|\alpha, \beta)$ in equation (3) using the Expectation-Maximization (EM) algorithm [30] to find the maximum likelihood estimate of β and α in the following equation (8) [28]. Based on the explanation of Griffiths & Steyvers [29], the calculation in equation (5) will involve the process of evaluating the probability distribution. Although $P(z|d)$ can be calculated for each z, but the

evaluation process will require too large a space. Therefore, we need a method in the form of sampling on the distribution of objectives using a Monte-Carlo Markov chain, sampling from a Markov chain that converges with the target distribution. Each value of the Markov chain is a value assignment for the sampled variable. By using Gibbs sampling, the following values will be obtained by sequentially sampling all variables from their respective distributions. In the LDA model using Gibbs sampling, the posterior distribution is obtained by equation (9) [29]

$$\ell(\alpha, \beta) = \sum_{m=1}^{M} \log P(d_m|\alpha, \beta) \tag{8}$$

$$P(z_i = k|z_{-i}, d) \propto \frac{n_{-i,k}^{(v)} + \beta n_{-i,k}^{(d_i)} + \alpha}{n_{-i,k}^{(\cdot)} + V\beta n_{-i,\cdot}^{(d_i)} + K\alpha} \tag{9}$$

where z_{-i} is a vector of the membership of the referenced topic without including the word w_i. The v index indicates that the word is the same as the v^{th} word in the vocabulary. $n_{-i,k}^{(v)}$ states how many times the v^{th} word of the vocabulary is grouped on the topic $k = z$ without involving the word w_i. The dot (.) indicates the addition operation is performed on the index. Meanwhile, d_i is a document containing the word w_i [28]. Furthermore, the values in ϕ and θ can be estimated based on the z values in any single sample using

$$\hat{\phi}_k^{(v)} = \frac{n_k^{(v)} + \beta}{n_k^{(\cdot)} + V\beta} \quad \text{and} \quad \hat{\theta}_k^{(d)} = \frac{n_k^{(d)} + \alpha}{n_k^{(\cdot)} + K\alpha}.$$

Next, the determination of the number of iterations is carried out through model formation experiments with several different k values chosen at random. For each model and each k value, perplexity is calculated and recorded, then plotted. From the plot, the trend of the value will be visually observed. The number of iterations chosen is the value or the earliest point at which a stable value trend begins to appear [12]. After determining the number of iterations, the LDA model is rebuilt to determine the candidate number of topics. The model development uses pre-selected multiple iteration inputs and is formed for several multi-topic candidates with a value in a certain range. From this model, the perplexity value of each candidate topic is obtained. Perplexity is a standard measure for topic modeling, which measures the model's ability to use training documents [12]. This method works by training several hidden variable models on two corpus of texts to compare the generalization performance of the model and the aim is to estimate the density in the hope of achieving a high likelihood value in the tests carried out. Perplexity monotonically decreases the likelihood value of the test data, and from an algebraic point of view, it is equivalent to the inverse of the geometric mean for the likelihood value of each word [11]. If d is a document, Q is stating the number of words in the document, w_q is q^{th} word in the d document and n is stating the number of words in the calculated sequence, then the probability value for the sequence of q words that appear together is written in the following equation

$$P(d) = P(w_1, w_2, \dots, w_q) = \prod_{i=1}^{Q} P(w_i|w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1}). \tag{10}$$

Equation (10) referred to as the n-gram model is used as the basis for calculating the likelihood value in equation (8) and also in the preparation of equation (3). The model in equation (10) is a statistical-based language model that is used to determine the probability value of word order appearing together from a training document [31]. For example, to get the probability value of $P(w_i|w_{i-2}, w_{i-1})$ in the trigram (3-gram) case, using the formula as shown in equation (11).

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{n(w_{i-2}, w_{i-1}, w_i)}{n(w_{i-2}, w_{i-1})}, \tag{11}$$

$n(w_{i-2}, w_{i-1})$ represents the number of consecutive occurrences w_{i-2}, w_{i-1} in the training data, and $n(w_{i-2}, w_{i-1}, w_i)$ is the number of consecutive occurrences w_{i-2}, w_{i-1}, w_i in the training data.

In order to measure the performance of a language model (model fitting), the theory of quantity of information or entropy is used [31]. This entropy value will be related to the perplexity value. H_p is the estimated entropy value that describes the probability that the model can meet the testing data. The lower the value of H_p, the resulting model is considered better. Furthermore, the perplexity value for a document using the n-gram language model approach is defined as equation (12) [31]. With some conditions and

assumptions in the LDA algorithm, in a test set involving a corpus D with M documents in it, the perplexity value of a collection of documents is obtained by modifying the above equation into this equation (13), where d_m declaring the m^{th} document into a corpus D; N_m is the number of words in each M document, and $\ell(\alpha, \beta)$ is the likelihood as in equation (8) [11].

$$perplexity = 2^{Hp} = 2^{\left(-\frac{1}{Q} log\, P(w_1, w_2, \ldots, w_q)\right)} \tag{13}$$

$$perplexity(D_{tes}) = \exp\left(-\frac{\sum_{m=1}^{M} \log P(d_m)}{\sum_{m=1}^{M} N_m}\right) = \exp\left(-\frac{\ell(\alpha,\beta)}{\sum_{m=1}^{M} N_m}\right) \tag{14}$$

The model is validated using 10-folds cross-validation to choose which candidate produces a good model, namely the model with the lowest relative perplexity value [21]. The 10-folds cross-validation is a validation method by breaking the training data into 10 parts and shuffling them at different positions (Figure 3 shows the illustration of the 5-folds cross-validation). This is done to reduce the possibility of the model overfitting (the model works too well) on the training data so that it has better performance when the model is applied to new data [7].

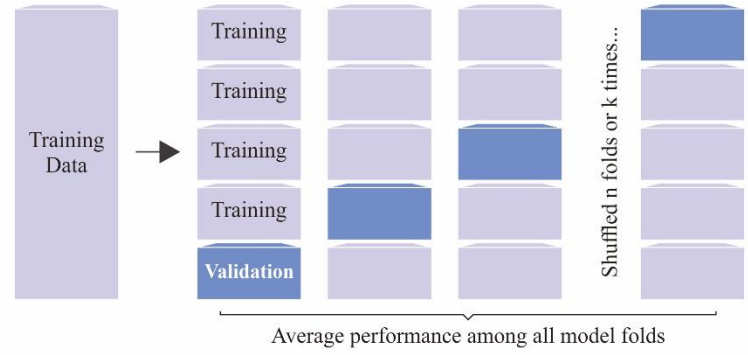

Figure 3. Illustration of cross-validation

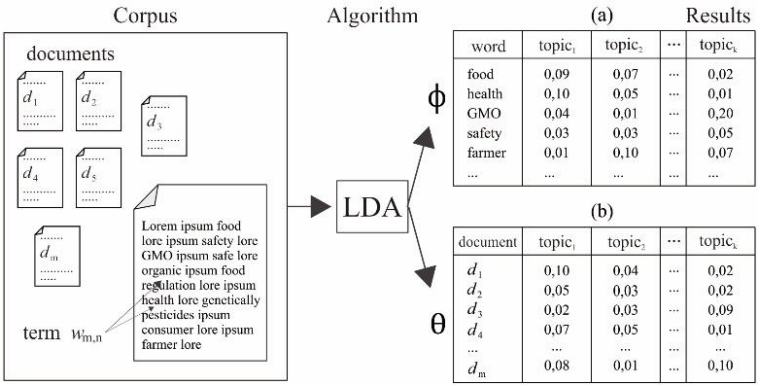

Figure 4. Illustration of matrix ϕ and θ

The process is continued by calculating the LDA model for the entire DTM by inputting the value of the number of iterations and topics selected in the previous process, and using the Gibbs sampling method. At this stage, the posterior distribution for the data will also be calculated. From the posterior distribution, the values of the word probability distribution on the topic will be obtained (matrix ϕ) and the proportion of topics for each document (matrix θ). The computational inference process is carried out to determine the membership of the document. From the modeling results obtained, further analysis will be carried out to reveal the words that make up a topic. By considering the probability distribution of each loaded word (matrix ϕ), each topic can be assigned a description such as a topic name or label. In addition, an analysis was also carried out to identify the membership status of each document against the resulting topic modeling, so that each document can be labeled according to the probability distribution of the emerging

topics (matrix θ). Furthermore, other descriptions can also be added to explain each processed data result. By conducting this analysis, recommendations and insights related to the modeling results will be obtained.

RESULT AND DISCUSSION

After the data has been obtained, the next step is to organize the data through the preprocessing stage. This stage includes removing punctuation, removing numbers, lowercase, lemmatization or stemming, removing stopwords, and stripping white space. The results obtained in this step can be seen for example in Table 1.

Table 1. Examples of preprocessing results

Step	Results
Original text	"It's a good app.....but.....one problem is that why can't we delete some photo for everyone??.....I mean we can delete messages for everyone but not some document or photo..... <U+0001F914><U+0001F47F>.....I am really angry with this app....."
Remove punctuation	"It s a good app but one problem is that why can t we delete some photo for everyone I mean we can delete messages for everyone but not some document or photo U 0001F914 U 0001F47F I am really angry with this app "
Remove number	"It s a good app but one problem is that why can t we delete some photo for everyone I mean we can delete messages for everyone but not some document or photo U F U FF I am really angry with this app "
Lowercase	"it s a good app but one problem is that why can t we delete some photo for everyone i mean we can delete messages for everyone but not some document or photo u f u ff i am really angry with this app "
Lemmatization	"it s a good app but one problem be that why can t we delete some photo for everyone i mean we can delete message for everyone but not some document or photo u f u ff i be really angry with this app"
Remove stopword	" delete photo delete message document photo angry "
Strip white space	"delete photo delete message document photo angry"

From the remove punctuation and remove number steps, every punctuation and number has been removed and replaced with a space. The next step is to convert each uppercase character to lowercase. Lemmatization or stemming aims to change each word into its basic form (lemma). In the remove stopword step, any words that do not have a special thematic meaning will be removed and replaced with spaces. Unnecessary spaces will then be removed in the white space strip step. A total of 1,710 documents that have passed the preprocessing stage (done with RStudio) will be converted into DTM. The DTM obtained are in the form of a 1710×2038 matrix with TF weighting. The weighting presents the term weight in each document as the number of occurrences of the term in a document. The description can be seen in Figure 5 (left). Each term registered in the DTM is unique, that is, no terms are the same in each column. The sparsity value is between 0 and 1, which in DTM is displayed as a percentage. The closer the value is to 1 (100%), the more terms appear in only a few documents (1 document, or even 0). Some of these terms are words that have typing errors. Because the term has no meaningful meaning, it will be removed with Remove Sparse Terms. The description of the repaired DTM can be seen in Figure 5 (right). The $1{,}260^{th}$ document only contains the word "somethings", so that the document is no longer listed in the repair DTM. As a result, the sparsity becomes 99%, with 1,145 terms are removed, leaving 893 terms and 1,709 documents in the DTM.

```
<<DocumentTermMatrix (documents: 1710, terms: 2038)>>
Non-/sparse entries: 12670/3472310
Sparsity           : 100%
Maximal term length: 21
Weighting          : term frequency (tf)
Sample             :
      Terms
Docs   call chat group message send status time update video whatsapp
  115     0    0     1       0    0      0    1      0     0        8
 1152     0    2     0       0    1      0    0      0     0        2
 1205     0    0     0       0    0      0    0      0     0        0
   13     0    1     0       1    1      0    0      0     1        0
 1469     2    3     0       0    0      0    0      0     1        2
  334     0    0     0       0    0      1    0      0     0        1
  589     3    0     0       2    0      0    0      0     1        1
   62     0    0     0       3    0      0    0      0     0        2
  849     0    0     0       0    0      3    0      0     0        0
  855     0    0     0       2    0      0    0      0     0        3
```

```
<<DocumentTermMatrix (documents: 1709, terms: 893)>>
Non-/sparse entries: 11525/1514612
Sparsity           : 99%
Maximal term length: 15
Weighting          : term frequency (tf)
Sample             :
      Terms
Docs   call chat group message send status time update video whatsapp
  115     0    0     1       0    0      0    1      0     0        8
 1152     0    2     0       0    1      0    0      0     0        2
 1159     4    0     0       0    0      0    0      0     4        0
    3     1    0     0       0    0      1    1      0     2        1
  318     1    0     0       0    0      0    0      0     0        0
  319     0    0     0       0    0      3    2      1     0        3
  587     0    0     3       1    0      2    2      0     1        0
  589     3    0     0       2    0      0    0      0     1        1
  593     0    2     2       0    0      0    0      0     3        1
  849     0    0     0       0    0      3    0      0     0        0
```

Figure 5. Description of DTM (left) and repaired DTM (right)

The DTM obtained from the previous step is then divided into two parts of data with a proportion of 50%, training data as many as 854 documents, and the remaining 855 documents as testing data. The candidate topics chosen for the model calculations are 15, 20, 25, 30, and 35. For each of these candidate topics, the perplexity value of the model is calculated in the iteration range from 1 to 1,500 in order to ensure clarity on a wide range of observations. Perplexity calculation is done by running the LDA algorithm in the topicmodels package. Training data is used as input for model making, then model fitting is carried out

with testing data input. With the device in use, this process takes 15 hours and 28 minutes (928 minutes). The determination of the number of iterations is done by observing the perplexity value. To facilitate the observation of value trends, a smooth line is drawn based on the distribution of perplexity values for each number of topics in Figure 6.

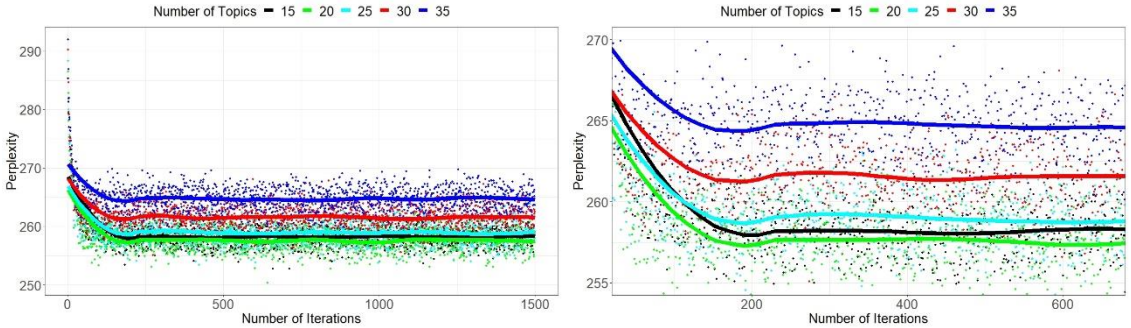

Figure 6. Smoothed line of perplexity value (left) and zoom (right)

In Figure 6, there are five candidate numbers of topics assigned different colors (15=black, 20=green, 25=cyan, 30=red, 35=blue). Each point represents the perplexity value for each model according to the candidate number of topics used. From these points, a smooth regression line is then made with a color that is adjusted to each model. On the left, it can be seen that the perplexity value is higher at the beginning of the graph, the more to the right (the larger number of iterations) the value decreases, and then starting at a certain point these values form a trend (stable pattern). The number of iterations selected is the earliest value at which a stable trend of values begins to appear in the overall topic. From Figure 6 it can be seen that the line begins to show a monotonous rate before the 500 iterations. Then, from the observation of the graph in Figure 6 (left) which is zoomed to Figure 6 (right), the number of iterations chosen is 300.

After the number of iterations are selected, we will continue with the determination number of topics. This stage uses two parts of data from DTM with proportions of 50%. Each of these data is training and testing data that has been used in the previous step. The calculation process through 10-fold cross-validation starts from breaking and randomizing the train set into 10 parts, followed by fitting the valid set. The number of iterations that have been entered is 300. The candidate topics (k) included in the calculation are from 2 to 100. This process produces a total of 990 records whose values will be visualized as graphs. Figure 7 shows a plot of the perplexity values obtained for each fold. For each candidate, the number of topics will have 10 points that represent the perplexity value of each fold. The blue line in the figure is a smooth regression line with respect to all points. After getting the calculation results, the plotting is again carried out to observe the perplexity value. From Figure 7 (left), it can be seen that the trend of perplexity values is relatively low in the range of 25 to 50. By observing the blue line by zooming Figure 7 (left) into Figure 7 (right), then choosing the number of topics is 31 (the lowest perplexity).

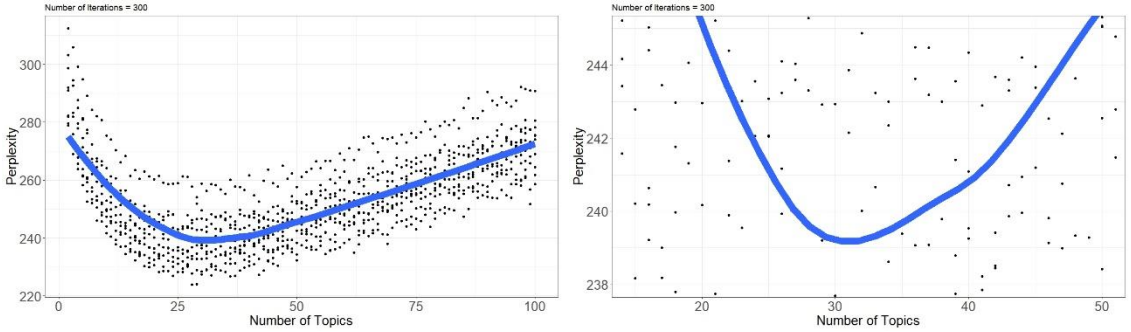

Figure 7. Cross-validation plot with 300 iterations (left) and zoomed (right)

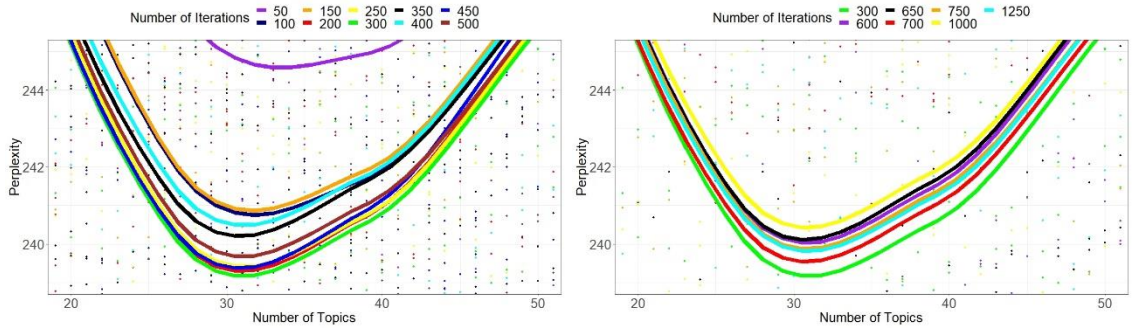

Figure 8. Perplexity of 50-500 iteration (left) and 300-1250 iterations (right)

To ensure that the selected number of iterations (300) is the optimum choice (can produce a model with the lowest perplexity value), a 10-fold cross-validation was also carried out on the other candidates for the number of iterations around 300, such 50, 100, 150, 200, 250, 350, 400, 450, 500, 600, 650, 700, 750, 1000, and 1250. The points are plotted and a smooth regression line is created for each candidate. Each number of iterations is plotted with a different color (300=green). In Figure 8, the left side shows a plot for the number of iterations 50, 100, 150, 200, 250, 300, 350, 400, and 500. From the figure it can be seen that the green line (300 iterations) shows the lowest perplexity value among the other lines. In Figure 8 on the right, the number of iterations is 300, 600, 650, 700, 750, 1000, and 1250. From the figure it is also found that the green line gives the lowest perplexity value compared to the other lines. The number of topics also shows that it remains at point 31. Many iterations that are not precise give a high perplexity value, such as for a model with 50 iterations. Thus, many iterations of 300 are the optimal choice for that model.

After determining the number of iterations and topics, the next step is to form a final model using these parameters. Modeling using the FitLdaModel function in the package textmineR. The modeling was run by entering parameters of 300 iterations and 31 topics (which were selected in the previous step). In the process of forming this model, two results that were obtained is a word-topic assignment (ϕ; phi) and *document-topic assignment* (θ; theta). (1) Word-topic assignment is one of the model results whose representation can be viewed as a matrix with the number of terms as the row size and the number of topics as the column size. For this result, a word-topic assignment of size 893×31; 893 terms and 31 topics. The sum of the probabilities of all words in each column is equal to 1. Each (i, k) entry in the word-topic assignment states the conditional probability of the i^{th} word to be included in the k^{th} topic. A word has a different probability on each topic. The greatest probability value in each line (for a word) states in which topic the word is most suitable. Meanwhile, the values in one column indicate the probability proportion of words that make up the topic model in that column. The greater the probability of a word in a column indicates that the word is more interpretive in explaining the latent characteristics of a topic compared to words with a smaller probability proportion. By sorting the proportion of the probability values of each word in each column (via Excel), the top terms with the greatest probability are obtained that can describe the characteristics of the related topic. (2) Document-topic assignment is similar to word-topic assignment, except that the line element that previously displayed the word is now replaced with a document number, so the document-topic assignment matrix is sized 1709×31; 1709 documents and 31 topics. Each entry in the i^{th} line of the document-topic assignment represents the conditional probability of each topic to appear in the document d. Meanwhile, each k^{th} column entry describes the proportion of the probability that a document can be in the topic k. The larger the proportion of a document, then the document is most suitable to be on the topic k. By taking the maximum probability of a document in one line (using Excel), a topic assignment for each document is obtained; the topic where the document is most suitable to be located. The documents with the highest probability on each topic are also collected as top documents. Furthermore, we will describe the characteristics of each modeling topic. One of the investigations carried out is to reobserve the proportion of top terms and top documents as a consideration to provide an intuitive description and interpretation of each topic. The labels on the 31 topics above are briefly listed in Table 2.

Table 2. Summary of topic labels

Topic	Label	Number of documents
1	Photo Delivery and Storage	68
2	Photo and Video Quality	100
3	Chat	46
4	Account Security	56
5	Upload Story Video	70
6	Upload Friend's Story	60
7	Online Status	54
8	Notifications	42
9	Audio and Video Calls	104
10	New Features	54
11	Call Quality	86
12	User Suggestions	47
13	Updates	51
14	Installation Problem	46
15	WhatsApp advantages	84
16	Voicemail	75
17	Group Chat	65
18	Feature Function	28
19	Data Backup	50
20	System Problem	39
21	Account Phone Number	37
22	User Satisfaction	77
23	Added Features	33
24	Application Security	24
25	Praise from Users	68
26	Media Download	50
27	Background Theme	30
28	Delete Message Feature	43
29	Ringtone	54
30	Storage Space	37
31	Animated Stickers	31

Interesting insights and recommendations are obtained from the table. Topics 5 and 6 can be used by the developer to determine the performance of the story feature. The reviews in topics 3, 16, 9, 1, and 11 can be used to evaluate how the calling and messaging feature performs. Voice and video (104 reviews), call quality (86 reviews), photos and videos (100 reviews), and voice messages were the most frequently discussed topics (75 reviews). The proportion of the maximum topic probability in the document is used to determine the topic assignment for a document. There is still a chance that the document will be assigned to another topic. Although it does not fully interpret the contents of the document, the maximum probability proportion can be considered the most likely topic based on the contents of the document. The proportion of top terms and top documents on each topic is used to label topics. This is done in order to obtain an overview and brief description of the subject. This can help developers to obtain information and recommendation more easily.

CONCLUSION

In this research, topic modeling has been carried out on user reviews of the WhatsApp application using Latent Dirichlet Allocation (LDA). The stages include: (1) The process of identifying the problem and taking the necessary data; (2) Preprocessing; (3) Determining the number of iterations through perplexity value investigation and selecting the number of topics with cross-validation. Determination of the number of iteration parameters by perplexity is a method that has not been widely used, where most studies give the number of iterations randomly; (4) Run the model using the number of iterations and the selected topic. This stage produces word-topic and document-topic assignments; (5) Furthermore, by observing several top terms and top documents on each topic, a description in the form of topic labels has been obtained that can interpret each topic intuitively. Top terms are used to determine the characteristics of the topic, then these findings are matched with the top documents in the topic. Broadly speaking, the topics covered in the data are calls, multimedia features, messaging, storage, and security issues. In this study, the number of topics is preceded by testing the parameters of the number of iterations through calculating perplexity, then the selected number of iterations will be used in determining the number of topics through cross-validation testing. In several other studies using LDA, the methods for determining the number of topics are quite diverse. Therefore, in future research, it is possible to compare the methods for determining the number of topics in LDA; which method yields the most interpretive topic. In addition, it can be added by measuring the accuracy level of the obtained topic modeling results.

REFERENCES

[1] Prasdika and B. Sugiantoro, "A review paper on big data and data mining: concept and techniques," *Int. J. Informatics Dev.*, vol. 7, no. 1, pp. 36–38, 2018.

[2] B. van der Sloot, D. Broeders, and E. Schrijvers, *Exploring the Boundaries of Big Data*. Amsterdam: WRR/Amsterdam University Press, The Hague, 2016.

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 1998.

[4] I. Feinerer, K. Hornik, and D. Meyer, "Text mining infrastructure in R," *J. Stat. Softw.*, vol. 25, no. 5, pp. 1–54, 2008.

[5] S. N. Asiyah and K. Fithriasari, "Klasifikasi berita online menggunakan metode support vector machine dan k-nearest neighbor," *J. Sains dan Seni ITS*, vol. 5, no. 2, pp. 2337–3520, 2016.

[6] I. Adiwijaya, "Text Mining dan Knowledge Discovery," 2006.

[7] T. Kwartler, *Text Mining in Pratice with R*. United Kingdom: John Wiley & Sons Ltd, 2017.

[8] B. Esmaeili, B. C. Wallace, H. Huang, and J. W. van de Meent, "Structured neural topic models for reviews," in *22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019, vol. 89.

[9] D. L. John *et al.*, "Topic modeling to extract information from nutraceutical product reviews," in *16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2019, pp. 1–6.

[10] R. Akila, S. Revathi, and G. Shreedevi, "Opinion mining on food services using topic modeling and machine learning algorithms," in *2020 6th International Conference on Advanced Computing and Communication Systems*, 2020, vol. 6, pp. 1071–1076.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[12] I. M. K. B. Putra, "Analisis topik informasi publik media sosial di surabaya menggunakan pemodelan latent dirichlet allocation (LDA)," Institut Teknologi Sepuluh Nopember, 2017.

[13] R. Annisa, I. Surjandari, and Zulkarnain, "Opinion mining on Mandalika hotel reviews using latent dirichlet allocation," *Procedia Comput. Sci.*, vol. 161, pp. 739–746, 2019.

[14] I. Sutherland, Y. Sim, S. K. Lee, J. Byun, and K. Kiatkawsin, "Topic modeling of online accommodation reviews via latent dirichlet allocation," *Sustain.*, vol. 12, no. 5, 2020.

[15] A. R. Destarani, I. Slamet, and S. Subanti, "Trend topic analysis using latent dirichlet allocation (LDA) (study case: Denpasar people's complaints online website)," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 5, no. 1, pp. 50–58, 2019.

[16] M. D. R. Wahyudi, A. Fatwanto, U. Kiftiyani, and M. G. Wonoseto, "Topic modeling of online media news titles during covid-19 emergency response in Indonesia using the latent dirichlet allocation (LDA) algorithm," *Telematika*, vol. 14, no. 2, pp. 101–111, 2021.

[17] M. Song, M. C. Kim, and Y. K. Jeong, "Analyzing the political landscape of 2012 Korean presidential election in twitter," *IEEE Intell. Syst.*, vol. 29, no. 2, pp. 18–26, 2014.

[18] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "iDoctor: personalized and professionalized medical recommendations based on hybrid matrix factorization," *Futur. Gener. Comput. Syst.*, vol. 66, pp. 30–35, 2017.

[19] X. Yang, D. Lo, L. Li, X. Xia, T. F. Bissyandé, and J. Klein, "Characterizing malicious android apps by mining topic-specific data flow signatures," *Inf. Softw. Technol.*, vol. 90, pp. 27–39, 2017.

[20] S. W. Putro, H. Semuel, and R. K. M. R. Brahmana, "Pengaruh kualitas layanan dan kualitas produk terhadap kepuasan pelanggan dan loyalitas konsumen restoran happy garden Surabaya," *J. Manaj. Pemasar.*, vol. 2, no. 1, pp. 1–9, 2014.

[21] G. M. Richardson, J. Bowers, A. J. Woodill, J. R. Barr, J. M. Gawron, and R. A. Levine, "Topic models: a tutorial with R," *Int. J. Semant. Comput.*, vol. 8, no. 1, pp. 85–98, 2014.

[22] E. L. Nylen and P. Wallisch, *Neural Data Science: A Primer with MATLAB® and Python™*. Academic Press, 2017.

[23] A. Bradley and R. J. E. James, "Web scraping using R," *Adv. Methods Pract. Psychol. Sci.*, vol. 2, no. 3, pp. 1–7, 2019.

[24] Nidhi, "Number of Topics for LDA on Poems from Elliston Poetry Archive," 2017. .

[25] H. Jelodar *et al.*, "Latent dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimed. Tools Appl.*, vol. 78, pp. 183–198, 2018.

[26] D. M. Blei, "Introduction to probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, 2011.

[27] Zulhanif, "Pemodelan topik dengan latent dirichlet allocation," 2016.

[28] B. Grün and K. Hornik, "Topicmodels: an R package for fitting topic models," *J. Stat. Softw.*, vol.

40, no. 13, pp. 1–30, 2011.
[29] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 101, pp. 5228–5235, 2004.
[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. R. Stat. Soc. Ser. B Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
[31] A. Juari and A. Purwarianti, “Deteksi OOV menggunakan hasil pengenalan suara otomatis untuk bahasa Indonesia,” *J. Ilmu Komput. dan Inf.*, vol. 2, no. 2, pp. 70–75, 2009.