# A TF-IDF Based Machine Learning Approach to Detect Emotion from Text

Vio Albert Ferdinand[1], Francis Alexander[2], Edwin Ario Abdiwijaya[3], Muhammad Amien Ibrahim[4]

*Computer Science Department, Bina Nusantara University*
*Jakarta, Indonesia 11480*

[1]vio.ferdinand@binus.ac.id

[2]francis.alexander@binus.ac.id

[3]edwin.abdiwijaya@binus.ac.id

[4]muhammad.ibrahim1@binus.edu

***Abstract*— Text-based communication is imbued with emotions. As technology advances, the need to detect emotions through text has become an urgent matter. Unfortunately, it can be challenging for machines to comprehend emotions in text-based communication. It is necessary to be cautious in deciphering the meaning because text can be interpreted in many ways. In this paper, we propose to solve the problem by applying a machine learning approach based on TF-IDF. First, the data are preprocessed using lemmatization and stop words. Then, features are extracted using TF-IDF and N-gram. Afterward, three machine learning models are implemented to classify the features: Multinomial Naive Bayes, Random Forest Classifier, and Linear SVM. Finally, the performance of each model is compared to find which model fits better for the problem. The results have shown that Linear SVM performs the best out of the three models with an accuracy of 89.50%, precision of 0.86, recall of 0.84, and F1-score of 0.85.**

***Keywords*—Emotion Detection, Natural Language Processing (NLP), Communication, TF-IDF, Machine Learning.**

## I. Introduction

Interaction and communication that takes place through writing, dialog, sign language, and other means is imbued with emotions and feelings [1]. Actors who are aware of emotion can decipher the genuine meaning of messages in their interactions. This also applies to interactive technologies, such as chatbots and social media. Social media is an interactive digital platform that can be used for various purposes, including serving as a place for two or more actors to communicate, disseminating the most recent information, and so on. Although most discussions nowadays are conducted mostly through images and videos, text-based conversation continues to play a vital role in communicating emotions [1].

As technology advances, many circumstances have established the need to detect emotions, e.g., dictating robots to act accordingly to human emotions [2], [3] and sentiment analysis for businesses [4]. Unfortunately, not all text-based dialogues are easy to comprehend, especially when the conservation is designed to portray emotions, and they might have a variety of ambiguous connotations at times [5]. Humans can distinguish these emotions, while machines require a similar level of comprehension and critical thought as humans [6]. Fortunately, machine learning models can assist them in identifying, distinguishing, and comprehending the meaning of distinct text-based dialogues.

This paper proposes a machine learning approach to detect emotion from the text. Additionally, we compare multiple machine learning models to see which ones provide better classification and prediction for emotion detection from text. In section II, we conduct a literature survey to identify and examine existing work in emotion recognition. We present a methodology to compare various learning models in section III. Implementations and all of the learning models' performance are discussed in section IV. We evaluate and conclude in section V.

## II. Literature Review

There are multiple approaches to identifying emotions from text data which will be discussed in the following sections.

### *A. Keyword-based Approach*

Compared to other methods, the keyword-based strategy is one of the most popular and intuitive. This method's main goal is to identify language patterns or essential characteristics that correspond to emotional categories. This can be done with lexicons such as Word-Net Affect and SentiwordNet. This dataset will go through more text preprocessing, including decontraction, stopword removal, tokenization, and lemmatization. It will then decide which emotion label is appropriate for each sentence in the dataset after going through all of that procedure based on some keyword dictionaries [7]. In these dictionaries, each word has a probability score for each emotion. Rahman et al. [8] defined 25 emotion classifications and presented a framework for sentence emotion recognition. It has an accuracy rate of 80% and is based on a set of proverbs, emoticons, keyword negation, short words, and keyword analysis.

### *B. Rule-based approach*

The rule-based technique is utilized to alter knowledge to examine information advantageously. It starts with text preprocessing, including stop word removal, POS tagging, tokenization, and lemmatization. Then, the rules of emotion are developed using the principles of statistics, language, and computing. After the rules are decided, the emotion labels are then determined by applying the rules to emotion datasets [9].

Dibyendu et.al [10] proposed a sentence emotion recognition technique by using semantic rules. This rule also includes negation words such as "not happy" and it obtained an F1 score of 66.18%. Badugu and Suhasini [11] also developed a Rule-Based Approach that can classify sentences from tweets with emotions and obtained an accuracy of 85%.

### *C. Machine Learning Approach*

Natural Language Processing (NLP), one of the branches of Artificial Intelligence (AI), uses numerous models to solve the classification problem of emotion detection from any sort of text. There are two types of Artificial Intelligence models: supervised learning and unsupervised learning. An annotated dataset of sentences and emotions will be utilized to train and test the classifier in supervised learning [9]. In comparison, the sentences will not be assigned various emotional labels during unsupervised learning. In [7], it is found that multiple studies have shown that supervised classification will yield results with higher accuracy than the unsupervised method.

### *D. Hybrid Approach*

The hybrid approach is a combination of multiple different approaches. This approach has proven to give the highest result in multiple studies rather than just using a particular method[12], [13]. This circumstance can occur because when implementing the approach, we can leverage the strength of one method we use and conceal the other's limitation in the process [9]. According to the previous surveys done on the comparison between the methods above, it is mentioned that the best individual method is the Keyword-based Method and Machine learning method. Although, it is essential to know that the one that receives the best accuracy is the hybrid method, combining two or more methods above [7]. As a result, this study proposes a solution that uses both Keyword-based Methods and Machine learning methods combined.

## III. Research Methodology

The emotion classification method can be divided into four main stages as shown in Fig 1, i.e. preprocessing, feature extraction, emotion classification, and performance evaluation.

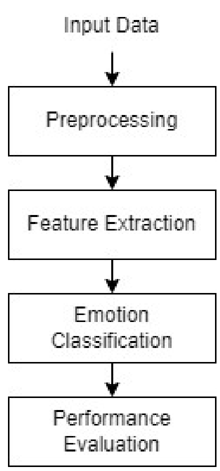


Fig. 1 Proposed Methodology for Emotion Classification

### *A. Preprocessing*

The preprocessing stage is done in four steps. Firstly, the data are cleaned. Then, the sentences are lemmatized. Finally, stop words are removed from the sentences. Details of all steps are explained below accordingly.

*1) Data Cleaning:* In this stage, data is cleaned to ease further processes. Firstly, duplicated rows are removed to discourage potential bias in the model. Afterward, the contractions in sentences such as "I'm" and "you're" are expanded. Then, words with no significant value to detect emotions, such as "http", "www", and others, are removed. The goal of cleaning the dataset is to ease the following processes and achieve a more optimal classification performance [9].

*2) Lemmatization:* Lemmatization is a process in which words are changed into their base form corresponding to the sentence structure, i.e. POS tag. This process is essential because it is undesirable to treat different word forms differently [14]. However, we should avoid doing so when the form represents different parts of the sentence structure. For example, "feel" and "feels", when both used in verb form, can be treated as one identical word, but it is not possible to do so if the word "feels" is used as a noun.

*3) Stop Words:* Stop words are a list of words that are excluded before the feature extraction process. Words that do not give valuable information towards the classification process can be problematic, especially during the classification step [9]. Hence, it is advised to remove such words to optimize the classification performance.

### *B. Feature Extraction*

From all available methods, the study uses TF-IDF to extract features from the sentences. The reason is that TF-IDF modeling has better performance when compared to other models such as Word2Vec [15]. TF-IDF weights the relevancy of a term to its document in a collection of documents [15]. In other words, TF-IDF measures the importance of a term in a document. Hence, it is possible to represent each document as a vector of TF-IDF values of words in a corpus. To acquire the values, TF and IDF are calculated beforehand.

TF is the relative frequency of a term to a document. To calculate this, we use the formula:

$$TF(t,d) = c(t,d) \,/\, |d| \tag{1}$$

In (1), c(t, d) is the count of the term t in document d and |d| is the number of terms in the document.

IDF measures how common a term is in a collection of documents. The calculations is done with the formula:

$$IDF(t, D) = log(N \,/\, f(t, D)) \tag{2}$$

In (2), N is the number of documents in the corpus and f(t, D) is the number of documents in corpus D which includes the term t.

After calculating both TF and IDF, TF-IDF is retrieved by multiplying the value of both measures. Specifically, the following formula is used to retrieve the TF-IDF of a term t to a document d in corpus D.

$$TFIDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (3)$$

In addition, n-gram is also employed during the TF-IDF process. Specifically, the study uses unigram and bigram when considering the terms in the TF-IDF process. In other words, each term during the TF-IDF calculation consists of either one word or two consecutive words. This is employed to allow the models to detect terms such as “not happy” and “never loved” better.

### C. *Emotion Classification*

Three different classification models are used to classify emotions from the feature vectors, namely Multinomial Naive Bayes (Multinomial NB), Random Forest Classifier (RFC), and Linear Support Vector Machine (SVM).

*1) Multinomial Naïve Bayes:* Multinomial NB uses a probabilistic approach to estimate how likely a class can occur given a set of features [16]. The model classifies the features by choosing the class that provides the largest probability value. The posterior probability of class y given feature vector x1 through xn is calculated using the formula:

$$P(y \mid x1,...,xn) = P(y) \times P(x1,...,xn \mid y) / P(x1,...,xn) \quad (4)$$

In (4), P(y) is the probability of class y, P(x1,...,xn | y) is the probability of the feature vector x1 through xn given class y, and P(x1,...,xn) is the probability of the aforementioned feature vector. Because the probability of a feature vector is constant given the input, it is possible to do classification based on the following proportional equation.

$$P(y \mid x1,...,xn) \propto P(y) \times P(x1,...,xn \mid y) \quad (5)$$

*2) Random Forest Classifier:* RFC is a classification method by generating numerous decision trees based on a set of randomly selected feature vectors [16]. By aggregating the results acquired in each decision tree, we can classify more reliably than using only one decision tree. Overall, random forest iterates the following steps to classify the given feature vectors.

- Create a bootstrapped dataset by randomly selecting feature vectors from the train dataset.
- Build a decision tree based on the bootstrapped dataset.
- Repeat step 1 and step 2 until satisfaction.
- Predict the feature vectors from the test dataset by aggregating the results from all decision trees.

*3) Linear Support Vector Machine:* SVM is a supervised learning model based on statistical learning, i.e. gradient descent [16]. The model depends on support vectors to generate the optimal position of the decision hyperplane that differentiate the classes. Hence, the model can predict unseen features more accurately.

While SVM supports various kernels, it is suggested that the linear kernel provides the most optimal performance. This result is because the feature vectors in NLP problems are discrete, sparse, and high dimensional so that the decision hyperplane tends to be linear[17], [18]. As a result, this study used Linear SVM as one of the classification models.

### D. *Performance Evaluation*

To evaluate how well each classification model performed, four evaluation metrics are employed, i.e. accuracy, precision, recall, and F1-score. The equation used for each metrics are as follows:

$$Accuracy = (TP + TN) / N \quad (6)$$

$$Precision = TP / (TP + FP) \quad (7)$$

$$Recall = TP / (TP + FN) \quad (8)$$

$$F1\text{-}Score = (2 \times Precision \times Recall)/(Precision+Recall) \quad (9)$$

In equation (6)-(9), N is the number of testing samples used during the classification process. On the other hand, TP and FP are used to represent the number of True Positive and False Positive respectively, while TN and FN are used to represent the number of True Negative and False Negative respectively.

## IV. Application and Evaluation

This section gives information on how the proposed method is applied and evaluated as follows. Firstly, the preprocessing stage is explained to provide better clarity on how the process is done. Afterward, the feature extraction process is explained. Finally, the implementation classification models are explained, and the performance of each model is compared.

### *A.* Dataset

All experiments in this study have been tested using the emotion dataset retrieved by HuggingFace using CARER [19]. The data given is separated into two types of variables, sentences as independent variables and emotion labels as a dependent variable. The independent variable consists of English sentences with a maximum length of 300 words. The given sentences are labeled with emotion; this emotion will be the dependent variable. The labels consist of six unique emotions: anger, fear, joy, love, sadness, and surprise. There are a total of 20.000 rows of data contained in this dataset which are then separated into training data, validation data, and testing data. The training data contains 16.000 rows, while validation data and testing data both contain 2.000 rows. The three data have the same emotion label distribution.

### *B.* Application and Evaluation on Preprocessing

After retrieving the dataset, the preprocessing phase is employed to the whole dataset as follows. The phase consists of 3 steps: data cleaning, lemmatization, and stop words removal. In the data cleaning step, the dataset is firstly checked for duplicated rows. It is found that there is one duplicate row in the training data. The duplicate row is then removed. Next, contraction expansion and meaningless words removal are then applied to the data. Table I and Table II provides an overview on how the dataset looks originally and after the process.

TABLE I
ORIGINAL DATA FROM CARER DATASET

| Document | Sentence | Emotion Label |
|---|---|---|
| D1 | i didnt feel humiliated | sadness |
| D2 | i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake | sadness |
| D3 | im grabbing a minute to post i feel greedy wrong | anger |
| D4 | i can t say i feel all that sympathetic | love |
| D5 | i feel really angry sometimes because for the love of god havent we been through enough | anger |

TABLE II
SENTENCES AFTER DATA CLEANING

| Document | Sentence | Emotion Label |
|---|---|---|
| D1 | i did not feel humiliated | sadness |
| D2 | i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake | sadness |
| D3 | i am grabbing a minute to post i feel greedy wrong | anger |
| D4 | i can not say i feel all that sympathetic | love |
| D5 | i feel really angry sometimes because for the love of god have not we been through enough | anger |

In the lemmatization process, the study used WordNet Lemmatizer. In this study, the process is implemented using NLTK library [20], defined by the function WordNetLemmatizer(). For the function to work properly, POS tagging is applied beforehand. The tagset used is obtained from WordNet tagset as described in Table III. The tagged sentences are described in Table IV. Then, the tagged sentence is feeded into the lemmatizer. Table V describes some of the sentences after applying lemmatization.

TABLE III
WORDNET TAGSET

| POS Tag | POS Name | Example |
|---|---|---|
| n | Noun | i, love, greedy |
| v | Verb | do, go, feeling |
| a | Adjective | sympathetic. Angry, happy |
| r | Adverb | just, not, really |

TABLE IV
TAGGED SENTENCES

| Document | Sentence | Emotion Label |
|---|---|---|
| D1 | i/n did/v not/r feel/n humiliated/v | sadness |
| D2 | i/n can/n go/v from/n feeling/v so/r hopeless/n to/n so/r damned/v hopeful/n just/r from/n being/v around/n someone/n who/n cares/n and/n is/v awake/n | sadness |
| D3 | i/n am/v grabbing/v a/n minute/n to/n post/n i/n feel/n greedy/n wrong/a | anger |
| D4 | i/n can/n not/r say/v i/n feel/n all/n that/n sympathetic/a | love |
| D5 | i/n feel/n really/r angry/a sometimes/r because/n for/n the/n love/n of/n god/n have/v not/r we/n been/v through/n enough/r | anger |

TABLE V
SENTENCES AFTER LEMMATIZATION

| Document | Sentence | Emotion Label |
|---|---|---|
| D1 | i do not feel humiliate | sadness |
| D2 | i can go from feel so hopeless to so damn hopeful just from be around someone who care and be awake | sadness |
| D3 | i be grab a minute to post i feel greedy wrong | anger |
| D4 | i can not say i feel all that sympathetic | love |
| D5 | i feel really angry sometimes because for the love of god have not we be through enough | anger |

Finally, stop words are removed from the sentences. In this study, the list of stop words is acquired from English stopwords provided by the NLTK library [20]. Negation words such as “no”, “not”, and “nor” are excluded as they contribute to how an emotion is expressed in the sentence. The dataset after removing the stop words is shown in Table VI.

TABLE VI
SENTENCES AFTER STOP WORDS REMOVAL

| Document | Sentence | Emotion Label |
|---|---|---|
| D1 | not feel humiliate | sadness |
| D2 | go feel hopeless damn hopeful around someone care awake | sadness |
| D3 | grab minute post feel greedy wrong | anger |
| D4 | not say feel sympathetic | love |
| D5 | feel really angry sometimes love god not enough | anger |

### C. Application and Evaluation on Feature Extraction

After the preprocessing stage, features are extracted from the dataset using TF-IDF and n-gram. The overall process is implemented in this study using scikit-learn [21], which is defined by the function TfidfVectorizer(). The parameters used are described in Table VII. A total of 2649 features are extracted as a result.

TABLE VII
TFIDFVECTORIZER() PARAMETERS

| Parameter Name | Parameter Value |
|---|---|
| ngram_range | (1,2) |
| min_df | 10 |
| norm | ‘l2’ |
| analyzer | ‘word’ |
| sublinear_tf | True |

### D. Application and Evaluation on Emotion Classification

Three classification models are implemented in the emotion classification stage, i.e. Multinomial Naive Bayes, RFC, and Linear SVM. The classification models are implemented using scikit-learn [21], which are defined by the function MultinomialNB(), RandomForestClassifier(), and LinearSVM() with default parameters.

After the training data are fitted in each model, the performance is compared using validation data and testing data as shown in Table VIII and Table IX. It is found that Linear SVM performs better than the other two models with accuracy of 89.70% and 88.85% for validation data and testing data respectively. This further proves that features tend to be linear in NLP problems, causing the model to perform better [17], [18]. Based on the results, the hyperparameters of Linear SVM are then tuned using GridSearchCV implemented in scikit-learn [21]. It is found that using the parameters in Table X gives the best results among other values, with accuracy of 89.95% and 89.50% in validation data and testing data respectively. Hence, it can be concluded that Linear SVM performed the best in comparison to other models evaluated in this study.

TABLE VIII
CLASSIFICATION PERFORMANCE COMPARISON USING VALIDATION DATA

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Multinomial NB | 83.65% | 0.90 | 0.69 | 0.74 |
| RFC | 87.20% | 0.85 | 0.85 | 0.85 |
| Linear SVM (default) | 89.70% | 0.87 | 0.86 | 0.87 |
| Linear SVM (tuned) | 89.95% | 0.88 | 0.86 | 0.87 |

TABLE IX
CLASSIFICATION PERFORMANCE COMPARISON USING TESTING DATA

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Multinomial NB | 83.00% | 0.88 | 0.66 | 0.70 |
| RFC | 87.10% | 0.82 | 0.82 | 0.82 |
| Linear SVM (default) | 88.85% | 0.85 | 0.83 | 0.84 |
| Linear SVM (tuned) | 89.50% | 0.86 | 0.84 | 0.85 |

TABLE X
PARAMETERS OF LINEARSVM() AFTER HYPERPARAMETER TUNING

| Parameter Name | Parameter Value |
|---|---|
| loss | squared_hinge |
| dual | True |
| tol | 1e-4 |
| C | 0.5 |
| class_weight | None |
| max_iter | 1000 |

## V. CONCLUSIONS

This paper has proposed a machine learning approach to detect emotions through text-based communication and compared different machine learning models to decide which is more suitable for the problem. We proposed to preprocess the dataset using Lemmatization and Stop Words Removal and extract features using TF-IDF and n-gram. The result shows that Linear SVM outperforms Multinomial NB and RFC with an accuracy of 89.50%, precision of 0.86, recall of 0.84, and F1-score of 0.85. Hence, it can be concluded that the proposed methodology can detect emotions accurately. In future works, we would like to investigate whether other preprocessing methods are more suitable than the ones used in this paper. Furthermore, we also would like to use the proposed methodology on larger datasets.

## REFERENCES


[1] E. C. C. Kao, C. C. Liu, T. H. Yang, C. T. Hsieh, and V. W. Soo, “Towards text-based emotion detection: A survey and possible improvements,” *Proc. - 2009 Int. Conf. Inf. Manag. Eng. ICIME 2009*, pp. 70–74, 2009, doi: 10.1109/ICIME.2009.113.

[2] J. C. Castillo, Á. Castro-González, F. Alonso-Martín, A. Fernández-Caballero, and M. Á. Salichs, “Emotion detection and regulation from personal assistant robot in smart environment,” *Intell. Syst. Ref. Libr.*, vol. 132, pp. 179–195, 2018, doi: 10.1007/978-3-319-62530-0_10.

[3] L. Zheng, Q. Li, H. Ban, and S. Liu, “Speech emotion recognition based on convolution neural network combined with random forest,” *Proc. 30th Chinese Control Decis. Conf. CCDC 2018*, pp.

4143–4147, 2018, doi: 10.1109/CCDC.2018.8407844.

[4] S. Opoku Oppong, D. Asamoah, E. Ofori Oppong, and D. Lamptey, "Business Decision Support System based on Sentiment Analysis," *Int. J. Inf. Eng. Electron. Bus.*, vol. 11, no. 1, pp. 36–49, 2019, doi: 10.5815/ijieeb.2019.01.05.

[5] N. Aldunate, M. Villena-González, F. Rojas-Thomas, V. López, and C. A. Bosman, "Mood detection in ambiguous messages: The interaction between text and emoticons," *Front. Psychol.*, vol. 9, no. APR, pp. 1–8, 2018, doi: 10.3389/fpsyg.2018.00423.

[6] A. Mikuckas, I. Mikuckiene, A. Venckauskas, E. Kazanavicius, R. Lukas, and I. Plauska, "Emotion recognition in human computer interaction systems," *Elektron. ir Elektrotechnika*, vol. 20, no. 10, pp. 51–56, 2014, doi: 10.5755/j01.eee.20.10.8878.

[7] K. Sailunaz and R. Alhajj, "Emotion and sentiment analysis from Twitter text," *J. Comput. Sci.*, vol. 36, 2019, doi: 10.1016/j.jocs.2019.05.009.

[8] T. Islam, R. R. Ema, and M. Humayan Ahmed, "Detecting Emotion from Text and Emoticon," *London J. Res. Comput. Sci. Technol.*, vol. 17, no. 3, pp. 9–13, 2017.

[9] A. R. Murthy and K. M. Anil Kumar, "A Review of Different Approaches for Detecting Emotion from Text," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1110, no. 1, p. 012009, 2021, doi: 10.1088/1757-899x/1110/1/012009.

[10] D. Seal, U. K. Roy, and R. Basak, *Sentence-Level Emotion Detection from Text Based on Semantic Rules*, vol. 933. Springer Singapore, 2020.

[11] S. Badugu and M. Suhasini, "Emotion Detection on Twitter Data using Knowledge Base Approach," *Int. J. Comput. Appl.*, vol. 162, no. 10, pp. 28–33, 2017, doi: 10.5120/ijca2017913366.

[12] H. Binali, C. Wu, and V. Potdar, "Computational Approaches for Emotion Detection in Text," *4th IEEE Int. Conf. Digit. Ecosyst. Technol. (IEEE DEST 2010*, 2010.

[13] S. P. Tiwari, M. Vijaya Raju, G. Phonsa, and D. K. Deepu, "A novel approach for detecting emotion in text," *Indian J. Sci. Technol.*, vol. 9, no. 29, 2016, doi: 10.17485/ijst/2016/v9i29/88211.

[14] K. Divya, B. Siddhartha, N. Niveditha, and B. Divya, "An Interpretation of Lemmatization and Stemming in Natural Language Processing," *J. Univ. Shanghai Sci. Technol.*, vol. 22, no. 10, pp. 350–357, 2020.

[15] D. E. Cahyani and I. Patasik, "Performance comparison of tf-idf and word2vec models for emotion text classification," *Bull. Electr. Eng. Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021, doi: 10.11591/eei.v10i5.3157.

[16] A. Nayak, "Comparative study of Naïve Bayes , Support Vector Machine and Random Forest Classifiers in Sentiment Analysis of Twitter feeds," *Int. J. Adv. Stud. Comput. Sci. Eng.*, vol. 5, no. 1, pp. 14–17, 2016.

[17] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," 1998, doi: 10.1515/9780691186740-014.

[18] N. Kalcheva, M. Karova, and I. Penev, "Comparison of the accuracy of SVM kemel functions in text classification," *Proc. Int. Conf. Biomed. Innov. Appl. BIA 2020*, pp. 141–145, 2020, doi: 10.1109/BIA50171.2020.9244278.

[19] E. Saravia, H. C. Toby Liu, Y. H. Huang, J. Wu, and Y. S. Chen, "Carer: Contextualized affect representations for emotion recognition," *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, pp. 3687–3697, 2018, doi: 10.18653/v1/d18-1404.

[20] S. Bird, E. Loper, and E. Klein, *Natural Language Processing with Python*. 2019.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research," *J. ofMachine Learn. Res. 12*, no. 9, pp. 2825–2830, 2011, doi: 10.1289/EHP4713.