

# Chatbot-Based Movie Recommender System Using POS Tagging

Muhammad Alwi Nugraha, Z K A Baizal\*, Donni Richasdy

School of Computing, Telkom University, Bandung, Indonesia

Email: <sup>1</sup>muhammadalwi@students.telkomuniversity.ac.id, <sup>2,\*</sup>baizal@telkomuniversity.ac.id, <sup>3</sup>donnir@telkomuniversity.ac.id

Correspondence Author Email: baizal@telkomuniversity.ac.id

Submitted:21/07/2022; Accepted:18/08/2022; Published: 30/09/2022

**Abstract**– The movie recommender system is a technology designed to make it easier for users to provide recommendations quickly and among the many pieces of information. Because the number of movies is huge, it causes a person to be confused in determining the choice of the movie to watch. Many movie recommending systems have been developed, but users cannot interact intensively. Based on these problems, we developed a chatbot-based conversational recommender system, which can interact intensively with the system. The developed chatbot uses normal language handling to permit the framework to comprehend what the user enters as natural language. POS Tagging is used to find tags in the form of movie titles with patterns in the POS Tagging model. However, the algorithm of those used on POS Tagging does not pay attention to the sentence entity, so the predicted title must correspond to the provisions of POS Tagging. Multinomial Naive Bayes looks for similarities of user input to datasets on intents. The dataset with the highest probability value or almost equal to the sentence entered by the user can be used as a response to user input. The test results of the chatbot application showed that the match rate between response and user input was 89.1%. Thus, the developed chatbot can be used well to provide practical and interactive movie recommendations.

**Keywords:** Chatbot; POS Tagging; NLP; Movies

## 1. INTRODUCTION

Technological advances are developing rapidly every day, accompanied by an increasing amount of information. Using the internet to search for information will become increasingly difficult because so much information is available. In the past, information could only be found in print media, and with the development of more advanced information technology, print media began to switch to electronic media. Nowadays, almost all the information needed can be obtained on the internet with various versions, and sometimes it confuses users because there is too much information available. The same happens with the pertinent information about the movie [1]. Due to the many movies available, users are confused when determining the choice to watch. Therefore, a system is needed that allows to receive information in accordance with the user's wishes. This framework is frequently called the recommender system [2]

The recommender system is a technology designed to make it easier for users to get recommendations quickly and efficiently against information overload [3]. The recommender system can give users an idea of what they might like based on the information obtained from that user [4], [5]. In this case, the movie is recommended based on other movies users like [6].

In the many movie recommender systems that have been developed, users are not given the opportunity to interact intensively with the system. In this research, a chatbot-based conversational recommending system was developed. The system allows users to have more interactions and conversations with it, similar to how they would talk with each other in daily life. The chatbot is a computer program used as a means for users to communicate with computers using natural language combined with artificial intelligence in order to automatically answer questions [7]. Therefore, users can interact write to the system in natural language, and it will process their input using Part of Speech Tagging method. POS Tagging performs grammatical identification of the text of the user [8]. The system will categorize words from sentences and look for tags related to the movie title in natural language. In this research, we evaluated the performance of the system in terms of user satisfaction by utilizing questionnaires [9].

There are several studies related to the topic of Chatbot recommender systems using natural language processing, research conducted by Saurav K, et al. in 2017 [10]. The research aims to build a medical chatbot that behaves like a virtual doctor that allows interaction with patients. Chatbots are developed in Python using pattern matching algorithms and natural language processing techniques. Based on a survey conducted to verify the performance of chatbots, 80% of the answers given by chatbots are correct answers, while 20% are ambiguous/incorrect answers from chatbots. According to the findings of this survey, chatbots can be used as virtual doctors for the education and teaching of nursing and medical students.

Lei Cui et al. in 2017 [11] explain that SuperAgent is a web-based chatbot that uses large amounts of data by utilizing NLP and machine learning techniques with Long short-term memory. The chatbot will provide a chat response with the information requested by the user. Otherwise, the information sought is not found, then the chatbot will respond that the data is not contained in the database. Meanwhile, Ma'arif, et al. in 2018 [12]. develop chatbots using the third-party application Line. In the research, it used PHP programming language as the main language and MySQL as the database. The limitations of this chatbot application are still linked to the Q&A mode that has been registered in the database so that interaction with users is limited to a predetermined vocabulary mode.

Endang S, et al. in 2017 [13] evaluate the formal part-of-speech Indonesian tagging used in the tweet tagging process Indonesian. This research was tested by adding five additional tags describing social media properties to an original tag set. The method of automated part-of-speech tagging is carried out through a hierarchical train process

with tweets annotated in 1000, 1600, and 1800. This shows that the process can achieve an accuracy of up to 66.36%. The experiment using the original label set had a slightly higher accuracy (67.39%), but missed the important information provided by the five additional labels.

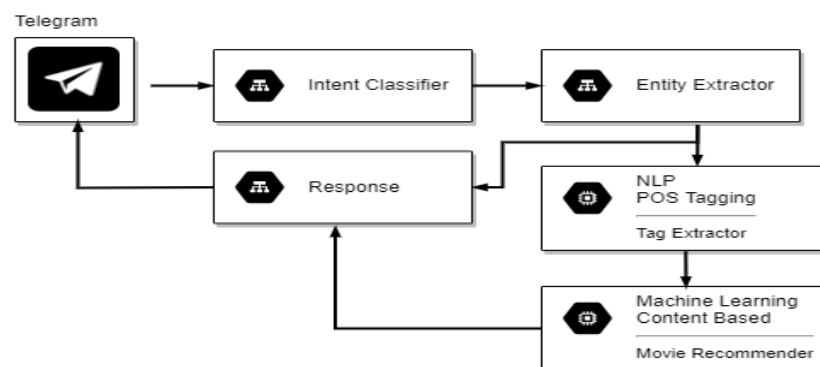
Warjri, S, et al. in 2019 [14] use the Hidden Markov Model to create the Khasi POS tagger. Some tags were used to create Khasi's tagset. Using a tagset, the corpus is manually tagged. They utilized up to 312 words in training and testing data, meanwhile the Khasi lexicon comprises around over 7,500 tokens. The HMM-based POS Tagger achieved a 76.70% success rate with the manually labeled Khasi vocabulary.

In earlier research, we utilized the pos-tagging approach to construct a chatbot for a movie recommendation system. In this research, we established a chatbot-based conversational movie recommendation system that enables users to interact with the system in a manner similar to a conventional chat. Our contribution to this research is that we first designed a chatbot system that begins with a review of the most recent research literature, creates documentation, and leads to the results of further research. Then we created a dataset of intents that will be used to train conversations on chatbots in this research, and finally, we began applying the chatbot system to conversations. This research consists of four parts, the first of which is an introduction that includes the research topic, related research, as well as the benefits and drawbacks of this research. Step 2 of the research technique includes the research phases and associated research phases. The third phase is the research's outcome, while the last step is the research's conclusion.

## 2. METHODOLOGICAL RESEARCH

### 2.1 Stages of Research

In this research, we developed a system that provides chatbot-based movie recommendations using tagging pos. Overall, the chatbot architecture is shown in Figure 1.



**Figure 1.** Chatbot Architecture

Based on the chatbot architecture shown in Figure 1, the first stage is intent classification. The entity extractor constitutes the second phase. The third stage is to identify each word in the phrase with tags that correspond to the word class, such as verbs, adverbs, NNP, and others, using the pos-tagging approach. At this point, the NNP tag is used to determine the movie's title. The next phase is to generate movie suggestions using a content-based algorithm based on the movie titles gathered in the previous step. The last phase is providing a response in the form of movie suggestion results in natural language.

### 2.2 Datasets of Intents

The Dataset of Intents is a collection of text data annotated with tags. Intents in this research used Indonesian, which will later be carried out Text Pre-processing. An example of a dataset can be seen in Table 1.

**Table 1.** Datasets from Intents

	text_input	Intents
0	Hi (Hi)	greeting
1	Hai (Hi)	greeting
2	Halo (Hello)	greeting
3	Apa kabar? (How Are You)	greeting
4	Selamat pagi (Good Morning)	greeting

### 2.3 Pre-processing

The preprocessing process is a stage to prepare questions as input data received by the system [10]. By removing unimportant elements from the data and making it easier for the model itself to understand the intent of the question before getting into NaiveBayes' Multinomial machine learning model. This preprocessing has the following stages:

- a. Case folding  
Case folding is carried out with the aim of converting all characters in the data to lowercase.
- b. Remove punctuations  
The Remove punctuations stage aims to remove any punctuation marks in the data.
- c. Tokenizing  
Tokenizing is the process of separating each word in a sentence [15]. In this case, the question that sent to the chatbot will be processed into a word/token

**2.4 Bag of words**

A Bag of Words (BoW) is a straightforward NLP and information retrieval model [16]. BoW model, independent of grammar or word order, depicts the text as a collection of words. The diversity of texts is preserved by the BoW approach. Counting the instances of each word in a text and using that number [17]. to describe the document with number so it can be understood by machine learning. The BoW method used in this research is the Count Vectorizer in scikit-learn. Table 2 is an example of a BoW using the Count Vectorizer method :

**Table 2.** Count Vectorizer Example

	ada	aja	aku	anda	apa
1	0	0	0	1	1
2	0	1	0	0	0
3	0	1	0	0	0
4	0	0	0	0	0
5	0	0	0	0	1

**2. 5 Multinomial Naive Bayes**

Multinomial Naive Bayes is a version of the Bayes algorithm that is used for classification. In this research, the Naive Bayes Multinomial Model was utilized for the classification of intents and the computation of probabilities for the prediction of intents. Through training with intent data, it is possible to determine a user's likelihood of occurrence (likelihood probability) if the relevant tag is provided. Multinomial naive bayes classifier also uses prior probabilities (prior probability) from a single kelas, which may be based on the anticipated applications of the data. According to the data analysis, if a new user submits a text including both words and numbers, the posterior probability of each tag with regard to the document in question may be established by comparing the posterior probability to the threshold. This approach considers not only whether words are included in the text, but also the frequency with which each term appears as follows (1,2) [18].

$$C_{MAP} = arg \max_{ce\{cl,cs\}} P(c|d) \tag{1}$$

$$C_{MAP} = arg \max_{ce\{cl,cs\}} P(c) \prod_{k=1}^m P(t_k|c) \tag{2}$$

The probability likelihood  $P(t_k|c)$  parameter is estimated as follows (3) by calculating events on all training materials in  $P(t_k|c)t_k c$  Laplacean prior is utilized to ensure that no test word has zero probability [19].

$$P(t_k|c) = \frac{1+N_k}{|V|+N} \tag{3}$$

which  $N_k$  is the number of instances of  $t_k$  in the training text in  $c$  and  $N$  represents the number of incidences of the term in  $c$ .

**2. 6 POS Tagging**

POS Tagging, often referred to as grammatical tagging, is the act of assigning a category to each word in a phrase or text. POS Tagging is an essential part of NLP, which translates human language into machine language. The general phases of the POS tagging process include the POS tagger analyzing sentences without class words and the tag set that impacts them so that words with class may be retrieved. POS Tagging is the process of labeling words or tokens as nouns, verbs, adjectives, etc. [20]. In this research, the purpose of using POS tagging was to improve the accuracy of values while enabling word search. Figure 2 shows the approach of POS tagging by labelling each word in a text..

```
[[('Berikan', 'VB'), ('saya', 'PRP'), ('rekomendasi', 'NN'), ('film', 'NN'), ('seperti', 'IN')
, ('Last', 'NNP'), ('Orders', 'NNP')]]
[('Last', 'NNP'), ('Orders', 'NNP')]
Last Orders
Last Orders
[INFO] Sukses Memberi Rekomendasi
```

**Figure 2.** Examples of annotating words

## 2. 7 Movie Recommender System

The movie recommendation system is a system that functions to recommend and predict movies that are in accordance with the user's wishes in order to help users support the selection of movies that meet their needs and desires. In the movie recommending system section using the Content Based method using datasets from Kaggle with a total of 4803 movie data and the process to be carried out is the data preprocessing process and the similarity measurement process using Cosine Similarity.

### a. Movie Datasets

In this research, the kaggle dataset was used, which contained 4803 movie titles and item content information in the form of titles, genres, and tags regarding movie items. These datasets are processed into datasets of movie titles and movie item content tags. The dataset to be used on the movie recommender system can be seen in Table 3.

Table 3. Sample data from a movie dataset

	movie_id	title	Tags
0	188927	star trek beyond	the uss enterpris crew explor the furthest rea...
1	10681	wall-e	wall· is the last robot left on an earth that ...
2	5174	rush hour 3	after an attempt assassin on ambassador han, i...
3	14161	2012	dr. adrian helmsley, part of a worldwide geophy...
4	17979	a christmas carol	miser ebenez scroog is awaken on christma eve ...

### b. Cosine Similarity

The *Cosine Similarity* process is used in the built movie recommendation system. It calculates the similarity between the text that the user enters and the text in the question data set [21]. The resulting value is the degree of similarity between the two texts. The similarity calculation is done using the sklearn library.

## 2. 8 Test Scenarios

The test scenario in this research used a beta test scenario with a user test survey directly using a questionnaire on satisfaction and evaluation of the application built [22]. against ten respondents.

# 3. RESULTS AND DISCUSSION

## 3.1 Preprocessing Result

The Preprocessing is the process of modifying text to prepare it for the next step of processing. In this step, Case Folding, Removal of punctuation, and Tokenization are performed. This operation is performed on the queries posed by the user. Table4 shows a process for Preprocessing.

Table 4. Preprocessing Result

	text_input	text_input_preprocessing
1	berikan saya rekomendasi film seperti Spectre?	berikan saya rekomendasi film seperti spectre
2	cara kamu bantu saya gimana?	cara kamu bantu saya gimana
3	bagaimana kamu bisa membantu saya?	bagaimana kamu bisa membantu saya

## 3.2 Pos Tagging Result

POS Annotation Tagging in this research only used Indonesian and searched for noun entities or NNP tags to search for specific movie titles. There are several classes of words used, for example, as in Table 5.

Table 5. POS Tagging examples

Before POS tagging	After POS tagging
Saya sebelumnya menonton film The Avengers di bioskop (I previously watched The Avengers movie in theaters)	[('saya', 'PRP'), ('sebelumnya', 'SC'), ('menonton', 'VB'), ('film', 'NN'), ('The', 'NNP'), ('Avengers', 'NNP'), ('di', 'IN'), ('bioskop', 'FW')]
Berikan saya rekomendasi film seperti Civil War (Give me a movie recommendation like Civil War)	[('berikan', 'VB'), ('saya', 'PRP'), ('rekomendasi', 'NN'), ('film', 'NN'), ('seperti', 'IN'), ('Civil', 'NNP'), ('War', 'NNP')]

## 3.3 Movie Recommender System

This research discussed the topic of a recommendation system using the content-based filtering approach, the CountVectorizer algorithm, and the cosine similarity method for determining the similarity of each item. In the process, the recommendation system should be able to give users with fresh references for other types of films that have commonalities with the films of interest to users. In addition, there is a difficulty with the system's ability to present movie suggestions while minimizing mistake rates.

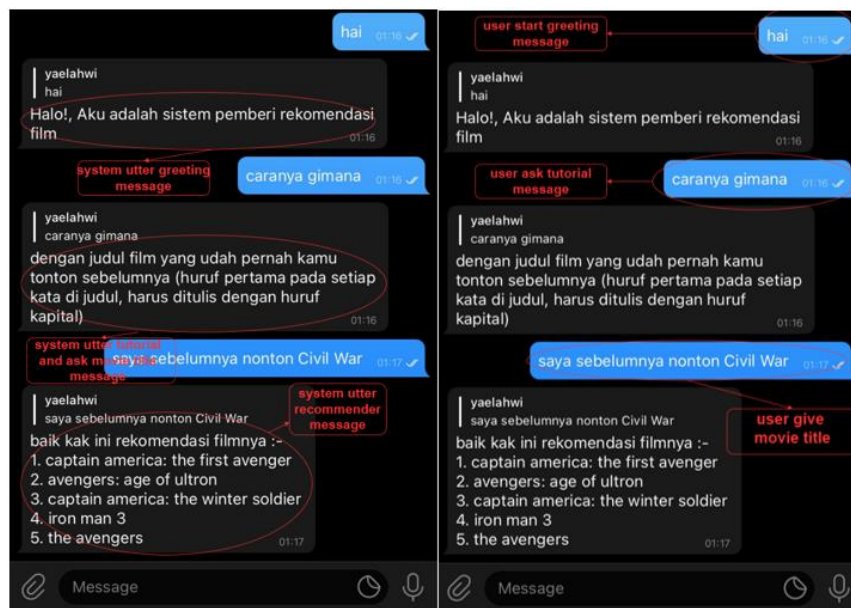


Figure 3. Recommendation Results

Based on the chatbot conversation shown in Figure 3, The user inputs natural language including the movie title object, and the chatbot provides movie suggestions based on past user input.

### 3.4 Test Results

In this research, Figure 4 demonstrates that the patterns are kept in a .json file. The pattern is processed, and the preprocessing outcomes are saved in a dataframe file.

```

{"tag": "bantu",
 "patterns": ["bagaimana kamu bisa membantu saya?", "cara kamu bantu saya gimana?", "gimana kamu bisa ngasih saya rekomendasi film?", "cara kamu",
 "responses": ["dengan cara kasih judul film yang sebelum nya pernah kamu nonton (huruf pertama pada setiap kata di judul, harus ditulis dengan
},
 {"tag": "tanya",
 "patterns": ["rekomendasi film", "saya pengen rekomendasi film", "tolong dong kasih rekomendasi film", "minta rekomendasi film", "aku minta rek",
 "responses": ["kamu sebelumnya nonton judul film apa (huruf pertama pada setiap kata di judul, harus ditulis dengan huruf kapital)", "kasih tau
},
    
```

Figure 4. Example of Pattern Stored in dataset Intent

Thus, when the user submits a phrase in the form of text data, the system may instantly utilize the preprocessed results of the intent data stored in the generated model. The results of the chatbot implementation are shown in Figure 5 as follows:

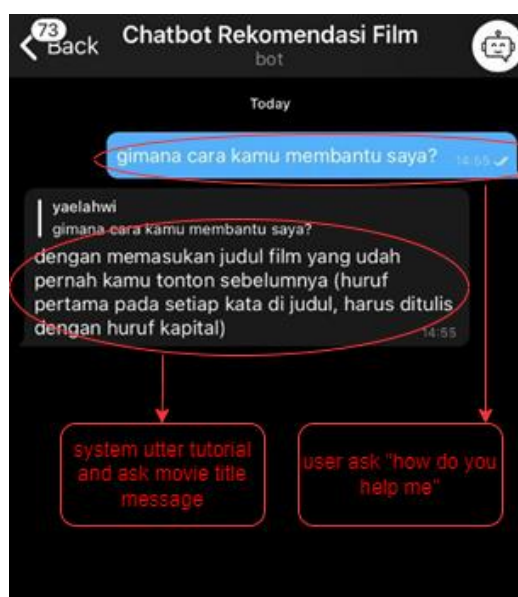


Figure 5. Chatbot Implementations



A total of ten people tested the method, and the results are shown in Figure 6. Errors were still identified in the system, according to the test findings:

- a. The user believes he has submitted a specific question, but the system responds to the inquiry choice in such a way that he is requested to choose one of the alternatives.
- b. Response falls short of user expectations.
- c. Due to the fact that the response to the user's inquiry is not yet added the system provides an inaccurate result.

### 3.5 Evaluation

Beta testing is an objective test carried out directly with users using a questionnaire about their satisfaction with the built application, as in Table 5. The test evaluation method used is a quantitative method based on user data. The Hquestionnaire assessment given to the user is calculated using a formula to determine the percentage of each answer by formula (4). Beta testing was conducted on chatbot users by asking questions about their experience with the chatbot. Chatbot testing is done directly by the user and provides objective feedback as well as using data from users to assess test results. The test results of each answer have been calculated using the percentage formula of the value tested on ten people. The results of the test assessment are in Table 6.

$$PS = \frac{VO}{MS} \times 100\% \tag{4}$$

PS = Percentage score

VS = Value Obtained

MS = Maximum Score

**Table 3.** Questionnaire Results Table

No.	Question	Scoring					Percentage (%)
		TS	KS	CS	S	SS	
1	Do users know what to do when opening a chatbot for the first time?	1	2		20	15	(38/50)*100%=76%
2	Are users familiar with chatbots?			6	4	35	(45/50)*100%=90%
3	Can the recommendation system chatbot give a good response?			3	16	25	(44/50)*100%=88%
4	Do users like the use of natural language when interacting with chatbots?				4	45	(50/50)*100%=100%
5	Do users enjoy interacting with chatbots?			3	8	35	(46/50)*100%=92%
6	Do users like the movie recommendations provided by the chatbot?				24	20	(44/50)*100%=88%
7	Are users helped by the chatbot application?				20	25	(45/50)*100%=90%
Total Scoring							(624/700)*100%=89.1%

The scale description in Table 7 is a scale used to measure the interpretation of values used to calculate the percentage of questionnaire answers

**Table 4.** Scale Description

Scale	Scale Description	Score
TS	Strongly Disagree	1
KS	Moderately Disagree	2
CS	Slightly Agree	3
S	Moderately Agree	4
SS	Strongly Agree	5

Resulting from input from users not being responded to in a manner that is consistent with the users expectations, For example, when there are a large number of patterns that all have the same probability value and weight, it causes the system to display "do not understand" to the user. Some pos tagger had sentence constructions that could not be processed by the grammatical rules recorded in the tagging pos, preventing the algorithm from accurately locating the title of the movie that matched the pattern. In addition to this, there is information provided by users, some of whose patterns do not appear in the purpose dataset. This may be due to the use of phrases that the system doesn't understand or slang words that aren't properly handled, resulting in a probability below the threshold. In order to evaluate the performance of the system, testing may be carried out by feeding the system up to twelve different user inputs at once. However, for user input with grammar that is not in compliance with Indonesian norms, it will be managed by removing certain tags to address mislabeling of words. As a result, The users will get the impression that they are interacting with other humans rather than a computer.

## 4. CONCLUSION



Based on the research findings of testing, It may be concluded that the constructed chatbot has an 89.1% response rate to user enquiries. The 10.9% disparity in results was caused by the restricted quantity of slang terms and the mix of diverse phrase structure patterns, which resulted in several searches producing the same near pattern value and failing to provide the desired result. Suggestions for future research, particularly with chatbot application development-related research, need to add more terms to the lexicon, including in Indonesian and slang dictionaries. In addition, current patterns in the knowledge base need to be expanded by merging diverse patterns and phrase forms in order to boost the probability of a better answer.

## REFERENCES

- [1] P. Vilakone, D.-S. Park, K. Xinchang, and F. Hao, “An Efficient movie recommendation algorithm based on improved k-clique,” *Human-centric Computing and Information Sciences*, vol. 8, no. 1, p. 38, Dec. 2018, doi: 10.1186/s13673-018-0161-6.
- [2] S. Khafid and I. Azis, “Perbedaan Disney+ Dan Netflix, Dari Biaya Berlangganan Hingga Fitur,” <https://tirto.id/perbedaan-disney-dan-netflix-dari-biaya-berlangganan-hingga-fitur-f4CS>., Sep. 17, 2020.
- [3] Z. K. A. Baizal, A. Iskandar, and E. Nasution, “Ontology-based recommendation involving consumer product reviews,” in 2016 4th International Conference on Information and Communication Technology (ICoICT), May 2016, pp. 1–6. doi: 10.1109/ICoICT.2016.7571890.
- [4] Z. K. A. Baizal, D. H. Widyantoro, and N. U. Maulidevi, “Computational model for generating interactions in conversational recommender system based on product functional requirements,” *Data & Knowledge Engineering*, vol. 128, p. 101813, Jul. 2020, doi: 10.1016/j.datak.2020.101813.
- [5] M. Polato and F. Aioli, “Boolean kernels for collaborative filtering in top-N item recommendation,” *Neurocomputing*, vol. 286, pp. 214–225, Apr. 2018, doi: 10.1016/j.neucom.2018.01.057.
- [6] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, “Content-Based Movie Recommendation System Using Genre Correlation,” 2019, pp. 391–397. doi: 10.1007/978-981-13-1927-3\_42.
- [7] A. Augello, G. Saccone, S. Gaglio, and G. Pilato, “Humorist Bot: Bringing Computational Humour in a Chat-Bot System,” in 2008 International Conference on Complex, Intelligent and Software Intensive Systems, 2008, pp. 703–708. doi: 10.1109/CISIS.2008.117.
- [8] E. M. Sibarani, Mhd. Nadial, E. Panggabean, and S. Meryana, “A Study of Parsing Process on Natural Language Processing in Bahasa Indonesia,” in 2013 IEEE 16th International Conference on Computational Science and Engineering, Dec. 2013, pp. 309–316. doi: 10.1109/CSE.2013.56.
- [9] S. Holmes, A. Moorhead, R. Bond, H. Zheng, V. Coates, and M. Mctear, “Usability testing of a healthcare chatbot: Can we use conventional methods to assess conversational user interfaces?,” in Proceedings of the 31st European Conference on Cognitive Ergonomics, Sep. 2019, pp. 207–214. doi: 10.1145/3335082.3335094.
- [10] S. K. Mishra, B. Dharendra, and N. Mishra, “Dr.Vdoc: A Medical Chatbot that Acts as a virtual Doctor,” *Journal of Medical Science and Technoslogy*, vol. 6, no. 3, 2017.
- [11] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, “SuperAgent: A Customer Service Chatbot for E-commerce Websites,” in Proceedings of ACL 2017, System Demonstrations, 2017, pp. 97–102. doi: 10.18653/v1/P17-4017.
- [12] D. P. Jati and M. R. Maarif, “The Development Of Chatbot Application On Line Messaging Platform For Customer Service In Jogja Sewa Kamera,” *Compiler*, vol. 7, no. 2, p. 91, Oct. 2018, doi: 10.28989/compiler.v7i2.368.
- [13] D. Munandar, E. Suryawati, D. Riswantini, A. F. Abka, R. Wijayanti, and A. Arisal, “POS-tagging for non-english tweets: An automatic approach: (Study in Bahasa Indonesia),” in 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), Nov. 2017, pp. 219–224. doi: 10.1109/ICICoS.2017.8276365.
- [14] S. Warjri, P. Pakray, S. Lyngdoh, and A. K. Maji, “Identification of POS Tag for Khasi Language based on Hidden Markov Model POS Tagger,” *Computación y Sistemas*, vol. 23, no. 3, Oct. 2019, doi: 10.13053/cys-23-3-3248.
- [15] A. Kaushal, L. White, M. Innes, and R. Kumar, “WordTokenizers.jl: Basic tools for tokenizing natural language in Julia,” *Journal of Open Source Software*, vol. 5, no. 46, p. 1956, Feb. 2020, doi: 10.21105/joss.01956.
- [16] D. Griol and Z. Callejas, “A Neural Network Approach to Intention Modeling for User-Adapted Conversational Agents,” *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–11, 2016, doi: 10.1155/2016/8402127.
- [17] S. Deepu, R. Pethuru, and S. Rajaraajeswari, “A Framework for Text Analytics using the Bag of Words (BoW) Model for Prediction,” *International Journal of Advanced Networking & Applications (IJANA)*, pp. 320–323, 2017.
- [18] A. S. Budi, “Klasifikasi Opini Green And Clean Kabupaten Lamongan Menggunakan Algoritma Multinomial Naive Bayes,” *Joutica*, vol. 3, no. 1, p. 125, Apr. 2018, doi: 10.30736/jti.v3i1.198.
- [19] D. H. Kalokasari, I. M. Shofi, and A. H. Setyaningrum, “Implementasi Algoritma Multinomial Naive Bayes Classifier Pada Sistem Klasifikasi Surat Keluar (Studi Kasus : Diskominfo Kabupaten Tangerang),” *Jurnal Teknik Informatika*, vol. 10, no. 2, Oct. 2017, doi: 10.15408/jti.v10i2.6199.
- [20] A. Yajnik, “General Regression Neural Network Based PoS Tagging for Nepali Text,” in *Computer Science & Information Technology*, Apr. 2018, pp. 35–40. doi: 10.5121/csit.2018.80603.
- [21] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, “Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms.”, 2021. doi: 10.5220/001072750000310.
- [22] R. A. Sekarwati, A. Sururi, R. Rakhmat, M. Arifin, and A. Wibowo, “Survey of Chatbot Testing Methods on Social Media to Measure Accuracy,” *Sisfotenika*, vol. 11, no. 2, p. 172, Jul. 2021, doi: 10.30700/jst.v11i2.1099.