

Indonesian part of speech tagging using maximum entropy markov model on Indonesian manually tagged corpus

Denis Eka Cahyani¹, Winda Mustikaningtyas²

¹Department of Mathematics, Universitas Negeri Malang, Malang, Indonesia

²Department of Informatics, Universitas Sebelas Maret, Solo, Indonesia

Article Info

Article history:

Received May 30, 2021

Revised Nov 5, 2021

Accepted Dec 27, 2021

Keywords:

Bigram

Maximum entropy markov model

N-gram

Part of speech tagging

Trigram

ABSTRACT

This research discusses the development of a part of speech (POS) tagging system to solve the problem of word ambiguity. This paper presents a new method, namely maximum entropy markov model (MEMM) to solve word ambiguity on the Indonesian dataset. A manually labeled "Indonesian manually tagged corpus" was used as data. Furthermore, the corpus is processed using the entropy formula to obtain the weight of the value of the word being searched for, then calculating it into the MEMM Bigram and MEMM Trigram algorithms with the previously obtained rules to determine the part of speech (POS) tag that has the highest probability. The results obtained show POS tagging using the MEMM method has advantages over the methods used previously which used the same data. This paper improves a performance evaluation of research previously. The resulting average accuracy is 83.04% for the MEMM Bigram algorithm and 86.66% for the MEMM Trigram. The MEMM Trigram algorithm is better than the MEMM Bigram algorithm.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Denis Eka Cahyani

Department of Mathematics, Universitas Negeri Malang

Jl. Semarang No.5, Sumber Sari, Lowokwaru, Malang, Jawa Timur 65145, Indonesia

Email: denis.eka.cahyani.fmipa@um.ac.id

1. INTRODUCTION

Recently, paying attention to writing sentence patterns is very important to do. Writing sentence patterns following the rules of the Indonesian language will make it easier for the person reading the sentence to be accepted so that it will reduce the bad impact of misperceptions between people in understanding the meaning of the sentence. Therefore, the writing that is made is expected to be conveyed informatively and communicatively to the general public. This research will discuss the importance of doing part of speech (POS) tagging in Indonesian. POS tagging is the process of automatically giving word-class labels to a word in a sentence so that it can help in arranging sentences according to good sentence patterns [1], [2].

In building Indonesian POS tagger, there are problems related to word ambiguity [3]. The ambiguity of the word in question is when there is the same word but has a different POS tag depending on the context of the sentence. An example is "Bisa ular kobra bisa mematikan" or "Cobra's venom can be deadly". The three words "bisa" or "venom" or "can" in the sentence are considered homonyms because they have different meanings but the pronunciation and spelling are the same. The first "bisa" or "venom" is a type of noun, while the second "bisa" or "can" is a type of verb. The difference in word type labeling is a problem because it will affect the POS tagging results and word ambiguity [4]. Ambiguous words can make readers confused because this ambiguous word has a double meaning. So it is feared that it could cause

misunderstandings when reading the sentence [5]. Based on this background, it is important to do research related to solve the ambiguity problem of POS tagging in Indonesian language.

Research on POS tagging for Indonesian has been developed previously. Several studies related to POS tagging in Indonesian that has been carried out are using rule-based methods which produce an accuracy rate of 79% [6], conditional random field (CRF) produces an accuracy rate of 83.72% using the 10-fold cross-validation test on the corpus II [7], hidden markov model (HMM) Bigram-viterbi and HMM Trigram-viterbi produce accuracy rates of 77.56% and 61.67% [8]. The previous study has a large dataset of more than 250,000 tokens as in the study [8], but the accuracy performance results have not been optimal. The accuracy performance needs to be improved so that it can solve the ambiguity problem in the POS tagging.

Previous research related to POS tagging was also carried out using HMM which resulted an accuracy rate of 96.50% [9]. Then, bidirectional long short-term memory produces an accuracy rate of 96.92% [10]. Other research related to POS tagging that has been conducted using the deep neural network for Turkish produces an accuracy rate of 88.7% [11], deep learning for Nepali produces an accuracy of 99% [12], maximum entropy for English produces an accuracy of 96.6% [13], HMM for Azerbaijani language yields an accuracy of 90% [14], HMM and morphological rules for Myanmar language get 94% precision [15], and CRF and Bi-LSTM for the arabic tweets get 96.5% accuracy [16]. Previous studies generate optimal accuracy values, but still using small amounts of dataset to conduct research experiments, as in research [12] using 100,720 tokens and 4,325 sentences. In this developed research, a larger number of dataset are used, namely 256,682 tokens and 10,000 sentences. So that this study has the novelty of using a larger amount of data than previous studies.

Based on the problems and several related studies, this research was conducted using the maximum entropy markov model (MEMM) method. The main contribution of this paper is to present a new method to solve the ambiguity problem of POS tagging, namely MEMM using large datasets. This paper improves a performance evaluation of research previously. The MEMM method provides a good level of accuracy in handling POS tagging because it can handle complex problems [17].

MEMM is a graphical model used to combine markov chain and maximum entropy (ME) features. The ME method can overcome the weakness of the HMM method where HMM can only calculate the possible observation words conditioned on the tag. ME can complement the HMM method by estimating the distribution parameters used for the transition probability separately and the POS tagging process can be carried out more efficiently [1]. MEMM can calculate a single probability function in each state and then compare it with the previous word and the word that will be given a word class label [1].

The MEMM method has some advantages over the HMM method. The MEMM method is considered to be able to solve the multi-feature representation problem which is a problem in the HMM method. The MEMM offers increased freedom in selecting features to represent observations [18]. For example, HMM has not paid attention to word spelling in the completion of the POS tagging case, while the MEMM method has paid attention to word spelling in solving the POS tagging case. In contrast to HMM which assumes independence between features [19], MEMM does not assume independence between features. Therefore, MEMM makes it possible to define many correlated yet informative features [1].

This paper discusses POS tagging using MEMM in the Indonesian dataset. The POS tagging uses the Indonesian language dataset from "Indonesian manually tagged corpus" in previous research [20] to label ambiguous words and words as a whole. The dataset is an adaptation of the penn tree bank corpus which is widely used as a reference for English POS tagging research and the POS tag values have been annotated manually. The dataset has large amount of data, consists of 10,000 sentences with a total of 256,682 tokens.

2. RESEARCH METHOD

The development of Indonesian POS tagging using MEMM approach. The stages taken in Indonesian POS tagging research using the MEMM algorithm are preprocessing, implementation of the MEMM algorithm, evaluation and analysis. The explanation of each stage in this research is:

2.1. Preprocessing

The data used in this study were taken from Indonesian manually tagged corpus, where all the data is presented in the form of sentences that have been sorted from manual processes. The dataset used consists of 10,000 sentences with a total of 256,682 tokens and consists of 23 POS tags (such as noun-NN, proper noun-NNP, and VB). The distribution of tags in the dataset is not balanced, where the most tags are NN with 55,575 tokens and the least tag is UH with 29 tokens. The results from the preprocessing stage will be input in the word error detection process. The preprocessing stage that will be carried out in this study [21], [22]: i) filtering, is the process of removing tags and words; ii) tokenizing, is the process of cutting the input string based on each word that composes it; iii) case folding, is the process of converting the entire text in a

document into a standard form, namely in lowercase form so that only letters A to Z are accepted while characters other than letters are eliminated; iv) removing spaces (“ ”) in a phrase that has more than one word. Example: “*Pesta olahragga*” or “Games”, “*Rumah sakit*” or “Hospital”.

2.2. Implementation of the maximum entropy markov model algorithm

After the preprocessing stage, the data on the corpus is processed using the MEMM feature which is used to make contextual predictions. The MEMM is the most common form of classification of maximum entropy [23], [24]. Maximum entropy is defined as the average maximum information value for a set of events X with a uniform probability value distribution [25]. The application of the MEMM algorithm begins by providing text input to the system. The text is preprocessed, then each word in the sentence will look for the probability value of the word class against the word class of the previous word in the corpus. The calculation of probability begins by calculating the probability of the first word by looking at the previous word (start). The probability of the second to the last word will be calculated by looking at the previous word-class using (1) [1].

$$H(x) = - \sum_x P(x) \log_2 P(x) \quad (1)$$

Where, $H(x)$ is entropy's value on variable X , $P(x)$ is the value of $\frac{Y}{x}$, x : the all words that appear in the sentence, $\log_2 P(x)$: formulated with the basic logarithm $\frac{\log x}{\log 2}$.

The difference between the implementation of MEMM Bigram and MEMM Trigram is if the MEMM Bigram is observed for the previous 1 tag, while MEMM Trigram is observed for the previous 2 tags [26]. Markov chain in applying the MEMM method serves to calculate the probability of an observable sequence of events [27]. If the word weight is known, then the MEMM calculation is carried out using (2). The output results obtained in this process are words and word classes by finding the best probability tags using (3) [1]:

$$p(c|x) = \frac{\exp(\sum_{i=0}^N w_{ci} f_i(c, x))}{\sum_{c' \in C} \exp(\sum_{i=0}^N w_{c'i} f_i(c', x))} \quad (2)$$

$$p(c|x) = \operatorname{argmax}_{c' \in C} \frac{\exp(\sum_{i=0}^N w_{ci} f_i(c, x))}{\sum_{c' \in C} \exp(\sum_{i=0}^N w_{c'i} f_i(c', x))} \quad (3)$$

Where:

e : 2,7

c : The word class of the designated data

x : Words from the entire dataset

c' : Entire class of predefined words

$f_i(c, x)$: Feature i for a particular class c for a given observation x .

w : Weighted word value

i : Observation word index

After knowing the best probability for each word in the sentence, the calculation of perplexity is a measure of the performance of language modeling based on word probability in the corpus. Perplexity is applied in this research as a form of validation against the comparison of the accuracy results obtained from the MEMM Bigram and MEMM Trigram methods. Perplexity generated by normalizing the testing data based on the number of words, meaning that minimizing perplexity is the same as maximizing probability. To calculate the perplexity in each sentence, the calculation for the testing data $W = w_1 w_2 \dots w_N$ can be done using (4) [1]. Bigram perplexity can be calculated using (5). And the Trigram perplexity can be calculated using (6) [1].

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \quad (4)$$

Where, $PP(W)$ is perplexity to the sentence, P is probability, w are the occurrence of the word on the corpus, N is total words in testing data.

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i|w_{i-1})}} \quad (5)$$

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i|w_{i-2}w_{i-1})}} \quad (6)$$

Where, $PP(W)$ is perplexity to the sentence, P is probability, w are the occurrence of the word on the corpus, N is total words in testing data.

The calculation of the probability of occurring words on the corpus is added with Laplace smoothing to handle the probability value of 0 (zero). It can be written in (7) for Bigram and (8) for Trigram. Meanwhile, the perplexity in the whole sentence can be calculated using (s_1, s_2, \dots, s_m) which is part of the corpus can be seen in (9) [1].

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + 1}{C(w_{i-1}) + V} \quad (7)$$

$$P(w_i|w_{i-1}w_{i-2}) = \frac{C(w_{i-2}w_{i-1}, w_i) + 1}{C(w_{i-2}w_{i-1}) + V} \quad (8)$$

Where, C is number of words, w_i are the designated word, w_{i-1} are the occurrence of 1 previous word in the corpus, w_{i-2} are the occurrence of 2 previous words in the corpus, V is total words in the training data (the same word counts as 1).

$$PP(C) = \sqrt[N]{\frac{1}{PP(s_1s_2 \dots s_m)}} \quad (9)$$

Where, $PP(C)$ is perplexity corpus, s are the perplexity result of each sentence, N is total words in testing data, m are the whole sentence on the data testing.

2.3. Evaluation and analysis

The evaluation carried out in this study consisted of three scenarios. The first evaluation was carried out to determine the level of accuracy of all words using the 10-fold cross-validation scenario. The second evaluation was carried out to determine the level of accuracy of the whole word using artificial testing data outside the corpus. The third evaluation is carried out to determine the level of accuracy on ambiguous words that are predicted to be correct.

In the first evaluation, the research was carried out applying the 10-fold cross-validation testing technique [28]. The dataset is divided into 10 parts, one part is used for testing and 9 parts are used for modeling (training) [29]. This means that when there is a corpus of 10,000 sentences of data, 1,000 sentences are used as testing data and 9,000 other sentences are used as training data [30]. Then, the second evaluation was done by dividing 1,000 sentences used as testing data (data outside the corpus) and 10,000 sentences used as training data. Artificial data outside the corpus used in the second test was obtained by manually collecting tokens in the Indonesian manually tagged corpus dataset. The third evaluation was done by calculating the number of ambiguous words in the corpus from the calculation results of MEMM Bigram and MEMM Trigram for the number of ambiguous words predicted accurately by the system based on the first evaluation scenario.

The accuracy of each result obtained from the 10-fold cross-validation is calculated by comparing the results with the original data. Accuracy can be obtained using (10) [31], [32]. Then the accuracy between MEMM Bigram and MEMM Trigram is compared so that it is known which method is better.

$$Accuracy = \frac{\text{number of data predicted correctly}}{\text{number of data predicted}} \quad (10)$$

3. RESULTS AND DISCUSSION

The results of applying the MEMM Bigram and MEMM Trigram algorithms are: From the test scenario with 10-fold cross-validation in first evaluation, there are a total of 10 experiments. The results of the calculation of the accuracy and the average accuracy of all the words obtained can be seen in Table 1.

The results of applying the MEMM Bigram and MEMM Trigram algorithms in Table 1 can be concluded that the highest accuracy is obtained in scenario 5 with an accuracy value of 85.18% for the MEMM Bigram algorithm and scenario 6 of 89.13% for the MEMM Trigram algorithm. The resulting average accuracy in first evaluation is 83.04% for the MEMM Bigram algorithm and 86.66% for the MEMM Trigram, respectively. In the second evaluation, data sharing was carried out, namely 10,000 sentences of training data and 1000 artificial sentences outside the corpus. From the second evaluation, the accuracy results for all words were 93.85% using MEMM Bigram and 94.17% using MEMM Trigram.

Table 1. Calculation of accuracy and average accuracy of overall words

Scenario	MEMM Bigram accuracy	MEMM Trigram accuracy
1	81.40%	85.11%
2	84.50%	87.76%
3	84.16%	87.71%
4	84.34%	88.64%
5	85.18%	88.90%
6	85.15%	89.13%
7	81.51%	84.97%
8	80.65%	84.58%
9	82.93%	85.02%
10	80.55%	84.80%
Average of Accuracy	83.04%	86.66%

The results of the application of the MEMM Trigram algorithm show higher accuracy than the MEMM Bigram in both the first and second evaluations. This is proven using perplexity. Perplexity validation was carried out in this study by taking 10 testing data (data outside the corpus). The results of calculating the perplexity Bigram and Trigram for 10 data testing can be seen in Table 2.

The results in Table 2 are then calculated using (9) and the results are 0.194216429 for MEMM Bigram and 0.181184234 for MEMM Trigram. These results indicate that the perplexity on the Trigram is smaller than the perplexity on the Bigram. These result also proven that the accuracy results using the Trigram are better than the Bigram accuracy results in the building POS tagger using the MEMM algorithm.

Table 2. Perplexity results

No	Perplexity Bigram results	Perplexity Trigram results
1	47820.33741	63053.74758
2	72302.91253	136004.8056
3	106849.6051	129885.1291
4	186060.2143	225805.7507
5	134523.1939	134519.3956
6	58395.40241	99034.39428
7	47024.58565	99298.23232
8	30833.01764	78606.7157
9	41918.24686	96455.01441
10	75934.56894	111161.9007
Result	0.181184234	0.194216429

On the third evaluation, the system is successful in labeling all words in the corpus. In addition, ambiguous words in the corpus, which numbered 91,851 were also successfully labeled by the system. Of all the ambiguous words, there are some ambiguous words in the corpus which can be properly labeled. However, not all ambiguous words are labeled correctly, there are also ambiguous words that are incorrectly labeled. Table 3 is an example of ambiguous words contained in the corpus. Table 4 is the result of ambiguous words that were predicted correctly.

Table 4 is the result of the test for handling the problem of word ambiguity. The number of ambiguous words predicted accurately using the MEMM Bigram algorithm is 87,099 from 91,851 words, and using MEMM Trigram algorithm is 89,650 from 91,851 words. From the number of ambiguous words

predicted correctly, the accuracy results in third evaluation obtained are 94.83% using MEMM Bigram and 97.60% using MEMM Trigram.

Table 3. Examples of ambiguous words

Word	Tag	Total number
“Akan” or “will” or “for”	IN	1
	MD	1806
“Hingga” or “until” or “to”	IN	348
	SC	39
“Untuk” or “to” or “for”	IN	396
	NN	1
	NNP	14
	SC	1839

Table 4. Results of ambiguous words predict exactly

Scenario	Ambiguous words predicted by Bigram	Ambiguous words predicted by Trigram
1	8911	9189
2	9455	9725
3	9612	9983
4	9519	9828
5	8936	9313
6	8228	8464
7	7711	7826
8	8691	8854
9	8264	8476
10	7772	7992
Total	87099	89650

Based on the results in Table 4, of all the ambiguous words contained in the corpus, not all are labeled correctly. There are also ambiguous words that are incorrectly labeled. Ambiguous word labeling is of two types that is ambiguous word labeling with exact Bigram results and incorrect Trigrams and ambiguous word labeling with incorrect Bigram results and exact Trigrams. The example of sentences from ambiguous word labeling of the first type in the corpus contained in the 6905th sentence, namely there is an ambiguous word “*agama*” or “religion”. This word can be categorized as an noun (NN tag) or an proper noun (NNP tag). In the context of the sentence in the example of the 6905th sentence, the word “*agama*” has an NN tag. The ambiguous word “*agama*” can be predicted correctly because the highest probability is 0.14348388496879 for the NN tag based on the calculation using MEMM Bigram. Meanwhile, the ambiguous word “*agama*” is predicted to be incorrect because the highest probability is 0.16282123413471 which is owned by the NNP tag based on the results of calculations using the MEMM Trigram.

The example of sentences from ambiguous word labeling of the second type in the corpus contained in the 9325th sentence, which contains the ambiguous word “*informasi*” or “information”, which can be categorized as an NN tag or an NNP tag. In the context of the sentence in the example of 9325th sentence, the word “*informasi*” has an NN tag. The ambiguous word “*informasi*” cannot be predicted with accuracy because the highest probability of 0.18027414056131 is owned by the NNP tag based on the results of calculations using MEMM Bigram. Meanwhile, the ambiguous word “*informasi*” is predicted precisely because the highest probability is 0.2361516773282 owned by the NN tag based on the results of calculations using the MEMM Trigram.

For example namely the labeling of the ambiguous word Bigram which is predicted to be correct, while the Trigram is predicted to be incorrect and vice versa is one of the drawbacks of this study, there is no maximum entropy feature that is used specifically to distinguish NN and NNP tags so that the system is difficult to distinguish between the two tags. In addition, this is also because the tags in the dataset are imbalanced, such as the “NN” tag has a total of 55,575 words, and the “UH” tag has a small number of words, which is 29 words. This imbalanced dataset affects the inaccuracy problem the probability value obtained by each word based on the calculation of MEMM Bigram and MEMM Trigram.

The difference between NN and NNP tags is NN tag indicate nouns in general and the writing is not capitalized unless it is at the beginning of the sentence. Meanwhile, the NNP tag shows specific nouns and the writing uses capital letters [33]. For example, the application of the maximum entropy feature to differentiate NN and NNP tags is [1].

$$f_1(c, x) \begin{cases} 1 & \text{if } w_i \text{ is lower_case } C = NN \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(c, x) \begin{cases} 1 & \text{if } w_i \text{ is upper_first } C = NNP \\ 0 & \text{otherwise} \end{cases}$$

Another problem that has not been fully resolved is phrases that have two words (multi-words), for the example: “*pesta olahraga*” or “Games”. This study still uses the usual tokenization process, namely separating words in each sentence into separate tokens. Then the tokenization is continued by removing the spaces for each token to make multi-words into a single word. To solve this multi words problem, it should be necessary to add a multiword expressions (MWE) tokenizer. MWE tokenizer itself will separate data on documents that are predicted to be MWE into separate tokens, so there is no need to remove spaces in each token.

Evaluation of system performance in this study improves the result compared to previous studies. This study also has advantages in shaping the model. The modeling in this study is quite easy through the stochastic tagger method. The model of the stochastic tagger method can be obtained by conducting training data on training data. The model that is formed can be used immediately to test the testing data. Another advantage of doing POS tagging using the MEMM algorithm is that it offers a variety of multi-feature representations so that it can form various maximum entropy functions to get the best accuracy results. The results of accuracy using MEMM in first evaluation reached 83.04% (Bigram) and 86.66% (Trigram) better than previous studies which used the same data [20]. The previous study using HMM Bigram-viterbi and HMM Trigram-viterbi only produce accuracy rates of 77.56% and 61.67% [8] and another previous study using rule-based methods produce the accuracy rate of 79% [6]. It proves that POS tagging using MEMM has advantages over the methods used previously. In addition, previous studies [8] have not used the second and third evaluation scenarios as used in this study. This is the advantage of this research because it evaluates using artificial testing data outside the corpus and calculating the number of ambiguous words in the corpus predicted accurately.

This research also has advantages in using a large number of datasets, which consist of 10,000 sentences and 256,682 tokens. The number of datasets is greater than the dataset used in the study [12] which only uses 100,720 tokens and 4,325 sentences. But the accuracy produced in this study is still below previous research [12] because previous studies used deep learning methods and this research used machine learning methods. The use of deep learning methods will usually improve accuracy results better than machine learning methods. Therefore, the future research is expected to use deep learning methods for the development of POS tagging.

4. CONCLUSION

Based on the research, it can be concluded that the research carried out succeeded in building an Indonesian POS tagger called “Indonesian manually tagged corpus” using the MEMM Bgram algorithm and the MEMM Trigram algorithm. The Indonesian corpus used consists of 10,000 sentences that have been given the POS tag manually, then the corpus is processed using MEMM Bigram and MEMM Trigram to get the entropy value. From the research results obtained, it is generally proven that using the MEMM method has advantages over the methods used previously which used the same data. This paper improves a performance evaluation of research previously. The resulting average accuracy is 83.04% for the MEMM Bigram algorithm and 86.66% for the MEMM Trigram in first evaluation. Meanwhile, in the second evaluation using testing data outside the corpus, the results obtained accuracy of 93.85% using MEMM Bigram and 94.17% using MEMM Trigram. In the third evaluation for ambiguous words, the accuracy results are 94.83% using MEMM Bigram and 97.60% using MEMM Trigram. From the results obtained, it can generally be concluded that the Trigram MEMM algorithm is better than the Bigram MEMM algorithm because the Trigram MEMM perplexity value has a lower value than the perplexity value in the MEMM Bigram. The problem of inaccuracy in POS tagging in this paper is influenced by the probability value obtained by each word based on the calculation of MEMM Bigram and MEMM Trigram. The suggestion for future research is the utilization of deep learning methods for the development of POS tagging and build a maximum entropy function that affects the distribution of the number of tags in the corpus. Example of making the maximum entropy feature as a function of distinguishing NN and NNP tags. In addition, tokenizer development also needs to be done to deal with words with phrases of more than one word or multi-word problems. To solve the multi-word problem, it should be necessary to add a MWE tokenizer, because using the maximum entropy markov model only, the NN and NNP tags are still biased so that they

cannot be distinguished significantly. But using the maximum entropy markov model has solved the problem of HMM where HMM can only calculate the possible observation words conditioned on the tag.




REFERENCES

- [1] D. Jurafsky and J. H. Martin, "Speech and language processing: An introduction to natural language processing," *Speech and Language Processing an Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, pp. 1–18, 2001.
- [2] X. Xue and J. Zhang, "Part-of-speech tagging of building codes empowered by deep learning and transformational rules," *Advanced Engineering Informatics*, vol. 47, Art. no. 101235, Jan. 2021, doi: 10.1016/j.aei.2020.101235.
- [3] L. Settipalli, R. Vedantham, and A. Chopparapu, "Ambiguity level assessment for large corpus using Embedded POS tagger with DNFC parameter," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 7, pp. 507–511, 2018.
- [4] A. D. W. Khan, J. A. Nasir, T. Amjad, S. Arafat, N. Aljohani, and F. S. Alotaibi, "Urdu part of speech tagging using conditional random fields," *Language Resources and Evaluation*, vol. 53, pp. 331–362, 2019, doi: 10.1007/s10579-018-9439-6.
- [5] E. Alayaboosar, A. Moloodi, and M. Kouhestani, "Word sense disambiguation focusing on POS tag disambiguation in persian: A rule-based approach," *International Journal of Information Science and Management*, vol. 17, no. 2, pp. 119–134, 2019.
- [6] F. Rashel, A. Luthfi, A. Dinakaramani, and R. Manurung, "Building an Indonesian rule-based part-of-speech tagger," in *2014 International Conference on Asian Language Processing (IALP)*, Oct. 2014, pp. 70–73, doi: 10.1109/IALP.2014.6973521.
- [7] F. Pisceldo, M. Adriani, and R. Manurung, "Probabilistic part of speech Tagging for Bahasa Indonesia," in *Proceedings of the 3rd International MALINDO Workshop, collocated event ACL-IJCNLP*, 2009, May.
- [8] D. E. Cahyani and M. J. Vindiyanto, "Indonesian part of speech Tagging using hidden Markov model Ngram & Viterbi," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Nov. 2019, pp. 353–358, doi: 10.1109/ICITISEE48480.2019.9003989.
- [9] A. F. Wicaksono and A. Purwarianti, "HMM based part-of-speech tagger for Bahasa Indonesia. In fourth international MALINDO workshop, Jakarta," *4th International MALINDO (Malaysian-Indonesian Language) Workshop*, 2010.
- [10] D. Handrata, C. N. Purwanto, F. H. Chandra, J. Santoso, and Gunawan, "Part of speech Tagging for Indonesian language using bidirectional long short-term memory," *2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019*, no. August, pp. 85–88, 2019, doi: 10.1109/ICORIS.2019.8874871.
- [11] C. A. Bahecevan, E. Kutlu, and T. Yildiz, "Deep neural network architecture for part-of-speech Tagging for Turkish language," *UBMK 2018-3rd International Conference on Computer Science and Engineering*, pp. 235–238, 2018, doi: 10.1109/UBMK.2018.8566272.
- [12] G. Prabha, P. V. Jyothsna, K. K. Shahina, B. Premjith, and K. P. Soman, "A deep learning approach for part-of-speech Tagging in Nepali language," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2018, pp. 1132–1136, doi: 10.1109/ICACCI.2018.8554812.
- [13] A. Ratnaparkhi, "A maximum entropy model for part-of-speech Tagging," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 133–142, 1996.
- [14] S. Mammadov, S. Rustamov, A. Mustafali, Z. Sadigov, R. Mollayev, and Z. Mammadov, "Part-of-speech Tagging for Azerbaijani language," in *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, Oct. 2018, pp. 1–6, doi: 10.1109/ICAICT.2018.8747154.
- [15] D. L. Cing and K. M. Soe, "Improving accuracy of part-of-speech (POS) tagging using hidden markov model and morphological analysis for Myanmar Language," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, 2020, doi: 10.11591/ijece.v10i2.pp2023-2030.
- [16] W. AlKhawter and N. Al-Twairesh, "Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM," *Computer Speech and Language*, vol. 65, Art. no. 101138, Jan. 2021, doi: 10.1016/j.csl.2020.101138.
- [17] J. Xiao, X. Wang, and B. Liu, "The study of a nonstationary maximum entropy Markov model and its application on the pos-tagging task," *ACM Transactions on Asian Language Information Processing*, vol. 6, no. 2, Art. no. 7, Sep. 2007, doi: 10.1145/1282080.1282082.
- [18] A. Sharma and V. Yadav, "Approaches to part of speech tagging in Hindi language: A review," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 283–291, 2020.
- [19] M. Bezoui, "Speech recognition of Moroccan dialect using hidden Markov models," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 1, pp. 7–13, Mar. 2019, doi: 10.11591/ijai.v8.i1.pp7-13.
- [20] A. Dinakaramani, F. Rashel, A. Luthfi, and R. Manurung, "Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus," in *2014 International Conference on Asian Language Processing (IALP)*, Oct. 2014, pp. 66–69, doi: 10.1109/IALP.2014.6973519.
- [21] D. D. Palmer, *Handbook of natural language processing*. CRC Press, 2000.
- [22] D. Virmani and S. Taneja, "A text preprocessing approach for efficacious information retrieval," in *Advances in Intelligent Systems and Computing*, pp. 13–22, 2019.
- [23] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 591–598, 2020.
- [24] C. Lv, D. Pan, Y. Li, J. Li, and Z. Wang, "A novel Chinese entity relationship extraction method based on the bidirectional maximum entropy markov model," *Complexity*, vol. 2021, pp. 1–8, Jan. 2021, doi: 10.1155/2021/6610965.
- [25] D. MacKay, "Information theory, inference, and learning algorithms," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2544–2545, Oct. 2004, doi: 10.1109/TIT.2004.834752.
- [26] H. Walia, A. Rana, and V. Kansal, "Comparative analysis of different classifiers for case based model in Punjabi word sense disambiguation," *Investigacion Operacional*, vol. 41, no. 2, pp. 273–288, 2020.
- [27] H. Wang, H. Fei, Q. Yu, W. Zhao, J. Yan, and T. Hong, "A motifs-based maximum entropy markov model for realtime reliability prediction in system of systems," *Journal of Systems and Software*, vol. 151, pp. 180–193, May 2019, doi: 10.1016/j.jss.2019.02.023.
- [28] P. Refaiezadeh, L. Tang, and H. Liu, "Cross-Validation," in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, pp. 532–538, 2009.
- [29] H. Zhou, "Cross-validation and ROC," in *Learn Data Mining Through Excel*, Berkeley, CA: Apress, pp. 67–81, 2020.
- [30] K. Srinivasan, A. K. Cherukuri, D. R. Vincent, A. Garg, and B. Y. Chen, "An efficient implementation of artificial neural networks with K-fold cross-validation for process optimization," *Journal of Internet Technology*, vol. 20, no. 4, pp. 1213–1225,




- 2019.
- [31] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, "Part 1: Simple definition and calculation of accuracy, sensitivity and specificity," *Emergency (Tehran, Iran)*, vol. 3, no. 2, pp. 48–49, 2015.
- [32] R. Vulanović and T. Mosavi Miangah, "A comparison of the accuracy of parts-of-speech Tagging systems based on a mathematical model," *Journal of Quantitative Linguistics*, vol. 26, no. 3, pp. 256–265, Jul. 2019, doi: 10.1080/09296174.2018.1474517.
- [33] S. Raharjo, R. Wardoyo, and A. E. Putra, "Detecting proper nouns in indonesian-language translation of the quran using a guided method," *Journal of King Saud University-computer and Information Sciences*, vol. 32, no. 5, pp. 583–591, Jun. 2020, doi: 10.1016/j.jksuci.2018.06.009.

BIOGRAPHIES OF AUTHORS



Denis Eka Cahyani    holds a Bachelor of Computer Science (S. Kom.) in Computer Science, Master of Computer Science (M.Kom.) in Computer Science, besides several professional certificates and skills. She is currently lecturing with the department of Mathematics at Universitas Negeri Malang, Malang, Indonesia. She is a member of the Engineers and the Institute of Electrical and Electronics Engineers (IEEE) Indonesia Section. Her research areas of interest include Natural Language Processing, Artificial Intelligent and Data Science. She can be contacted at email: denis.eka.cahyani.fmipa@um.ac.id.



Winda Mustikaningtyas    holds a Bachelor of Informatics degree from Sebelas Maret University, Indonesia in 2020. She is a junior analyst at Grup Pengembangan Aplikasi Sistem Informasi OJK, Jakarta, Indonesia. The project currently being developed is the Non-Bank Financial Industry Supervision Information System in supervisory unit. She can be contacted at email: windaamustikaningtyas@gmail.com