Part of Speech Tagger for Indonesian Lannguage Using Conditional Random Field (CRF)

Fivip Saefulloh

Informatics Engineering – Indonesia Computer University
Jl. Dipatiukur 112-114 Bandung
Email : fvpsaefulloh@gmail.com

ABSTRAK

Part-of-speech tagging is the process of assigning a part-of-speech marker to each word in an input text [1]. In this study, the method used to develop POS Tagger is Conditional Random Field (CRF). CRF is one of the probabilistic methods used in the development of POS Tagging and has advantages among other probabilistic methods, such as Hidden Markov Model (HMM) and Maximum Entropy Methods [8,9]. For the training and testing data comes from "Indonesian_Manually_Tagged_Corpus" [2] contains 250000 words with 23 tagset. Data training that used in training process contain 3910 words and for testing used 4 file data testing with each file contain 50 sentences with different number of words for each testing data. From the testing results performed on average the resulting accuracy is 78,20%.

Kata Kunci : Tagging, CRF, Part of speech, Indonesian Language

1. INTRODUCTION

In Indonesian've done some research on POS tagging, that is : *Probabilistic Part Of Speech Tagging for Bahasa Indonesia* [3] using probabilistic methods Maximum Entropy Models with the highest accuracy up to 97,57% and Conditional Random Field with the highest accuracy up to 91,15%, *HMM Based Part-of-Speech Tagger for Bahasa Indonesia* [4] with accuracy 96,50% for corpus that contain 15% OOV Words and 80,4% for OOV words, *Pengembangan Part of Speech Tagger untuk Bahasa Indonesia Berdasarkan Metode Conditional Random Fields dan Transformation Based Learning* [5] with accuracy 80,21 % for CRF and *Building an Indonesian Rule-Based Part-of-Speech Tagger* [6] with the highest accuracy 79%.

In general, the algorithm or the method used in the development of POS Tagging is divided into three ways [1,7] : Rule-based Tagger, Stochastic Tagger dan Transformation-based Tagger.

Rule-based tagger done by top-down approach, that is with consulting with linguistic experts to determine the rules commonly used by humans.

Stochastic tagger is done by the bottom-up approach, using the corpus as training data to determine the best tag probabilistic for a word in a context.

Transformation-based tagger is a combination of both methods. Similarly, as well as rule-based tagger this way based on the rules that have been determined when seeing the words are ambiguous. This method also has a machine learning component to retrieve rules automatically from the training corpus.

In this research the method is a stochastic tagger using Conditional Random Field (CRF). In the Indonesian language itself has carried out research on the development of these methods with accuracy 91.15% and 80.21% [4,6] using 25 and 37 tagset using the corpus with 26.346 words. From these results, the accuracy is still not optimal. Based on this, it still needed further research to improve the accuracy by using this method. In this study, the corpus used is "Indonesian_Manually_Tagged_Corpus" [2] contains 250000 words with 23 tagset.

1.1 Word

According to Kamus Besar Bahasa Indonesia (KBBI), Word is an element of spoken or written language that embodies the unity of feelings and thoughts that can be used in the language.

1.2 Word Classes

In determining the word classes, a principle that needs to be held is the fact that the class of words or categories of words is part of the syntax, so the characteristics of each word must be explained in terms of syntax. Here is the class division of words in Indonesian [13]:

Table 1 Indonesian Word Classes

No	Kelas Kata
1	Verb
2	Adjective
3	Noun
4	Pronoun
5	Number
6	Adverb
7	Interrogative
8	Demonstrative
9	Article
10	Preposition
11	Conjunction
12	Phatic Category
13	Interjection

1.3 Conditional Random Field

CRF is a framework to build a probabilistic model for segmenting and labeling sequence data. CRF main advantages compared with the hidden Markov model (HMM) is the conditional nature, thus reducing independence assumptions required by HMM to ensure inference tractable [8]. Besides CRF also avoids bias label problems, the weakness shown by maximum entropy Markov model (MEMM) and other models of conditional Markov that is directed graphical models [9].

Introduced by Lafferty et al (2001) [8], Conditional Random Field (CRF) is probabilistic model to compute probabilty $p(y|x)$ from possible output $y = (y_1, y_2, \dots, y_n)$ given input $x = (x_1, x_2, \dots, x_n)$ that called observation data [10]. Conditional probability y given x can be written [11] :

$$P(y|x) = \frac{1}{Z(x)} \prod_c \psi_c(y_c, x) \quad (1)$$

$$Z(x) = \sum_y \prod_c \psi_c(y_c, x) \quad (2)$$

Where $\psi_c(y_c, x)$ positive function or potensial function, and $Z(x)$ is partition function or normalization function. In the implementation there are two kind of potential function, that is node potential $\phi(y_t, x)$ and edge potential $\psi(y_t, y_{t+1}, x)$. Node potential can be notated to the following equation.

$$\phi_t(y_t, x) = \exp(\sum_k \lambda_k f_k(y_t, x, t)) \quad (3)$$

As for a potential edge can be written as follows [11].

$$\psi_t(y_t, y_{t+1}, x) = \exp(\sum_k \lambda_{k'} f_{k'}(y_t, y_{t+1}, x, t)) \quad (4)$$

Where :

x	: input data
y_t	: tag for data x
y_{t+1}	: tag after y_t
t	: indeks for word t from input data
λ_k	: node feature parameter
$\lambda_{k'}$	: edge feature parameter
$f_k(y_t, x, t)$	: node feature number k
$f_{k'}(y_t, y_{t+1}, t)$	: edge feature number k

In general, the steps in the conditional random field (CRF) method is as follows.

1.3.1 Feature Extraction

Features are the most important element to build this model. Selection of the right features can improve the accuracy of labeling. In this study there are two features, the node features and edge features. Node Features is association between the label y with the input data x, the following are examples of node feature :

$$f_t(y_t, x, t) \begin{cases} 1 \; if \; y_t = CC \;\; , x = Saya \\ 0 \quad otherwise \end{cases}$$

The feature value above will be 1 if usually word 'saya' has tagset CC, otherwise 0. While the edge feature is association between the part of speech for word being processed by the part of speech for word thereafter. Here is an example of edge features :

$$f_t(y_t, y_{t+1}, x, t) \begin{cases} 1 \; if \; y_t = NNP \; , y_{t+1} = VB \\ 0 \quad otherwise \end{cases}$$

The edge feature value will be 1 if label NNP usually follow by label VB, otherwise 0.

1.3.2 Training

The training process is done to obtain optimal values for the node features parameters and edge features parameter. The parameter values obtained by using maximum likelihood can be written :

$$\mathcal{L} = \sum_{t\in[1,T]} \sum_k \lambda_k f_k(y_t, x, t) - \sum_{t\in[1,T-1]} \sum_{k'} \lambda_{k'} f_{k'}(y_t, y_{t+1}, x, t) - \log Z(x) - \sum_k \frac{\lambda_k^2}{2\sigma^2} \quad (3)$$

Where :

$t \in [1, T]$	: data
λ_k , $\lambda_{k'}$	: feature parameter number k
$f_k(y_t, x, t)$, $f_{k'}(y_t, y_{t+1}, x, t)$	: feature number k
$Z(x)$	: partition function
σ	: standard deviation

Where $Z(x)$ is partition function with respect to the input x, and σ is the standard deviation of the Gaussian distribution. Last term is ro 'regularise' the objective function as it penalizes large parameter λ_k.

To get the parameter λ for each function features, then used the log-likelihood derivative, can be written [11]:

$$G_k^x = \sum_{t\in[1,T]}\left(f_k(y_t,x,t) - \sum_{y_t} P_t(y_t|x) f_k(y_t,x,t)\right) - \frac{\lambda_k}{\sigma^2} \quad (4)$$

$$G_{k'}^x = \sum_{t\in[1,T-1]}\left(f_{k'}(y_t,y_{t+1},x,t) - \sum_{y_t}\sum_{y_{t+1}} P_t(y_t,y_{t+1}|x) f_{k'}(y_t,y_{t+1},x,t)\right) - \frac{\lambda_{k'}}{\sigma^2} \quad (5)$$

Where :

G_k^x	: score of gradient number k for node feature
$G_{k'}^x$	: score of gradient number k for edge feature
$f_k(y_t,x,t)$	: node feature number k
$f_{k'}(y_t,y_{t+1},x,t)$	: edge feature number k
$P_t(y_t\|x)$	: the probability tag y_t over data data x
$P_t(y_t,y_{t+1}\|x)$	: the probability tag y_t followed by tag y_{t+1} when data x
λ_k	: node feature parameter number k
$\lambda_{k'}$	: edge feature parameter number k
σ	: standard deviation

Equation 6 and 7 above is stochastic gradient method. To calculate $P_t(y_t|x)$ and $P_t(y_t,y_{t+1}|x)$ used *two-pass* procedure through data. This procedure is well known as *forward-backward.*

1. Forward Pass

Forward pass did a search from index t = 1 to t = T. To calculate the local probability required search across in the opposite direction (backward pass).

Variable forward denoted by $\alpha_t[y_t]$. When t = 1 initialization $\alpha_1[y_1] = 1/S$ for the entire label y_t, where S is the total number of word class label. When t ≥2 it can be used the following equation [11]:

$$\alpha_t[y_t] = k_t \sum_{x_{t-1}} \alpha_{t-1}[y_{t-1}]\phi_t(y_t,x)\psi_{t-1}(y_{t-1},y_t,x) \quad (8)$$

Where :

$\alpha_t[y_t]$	: forward pass variable number t
k_t	: scaling factor to ensure $\sum_{y_t}\alpha_t[y_t] = 1$
$\alpha_{t-1}[y_{t-1}]$	: previous forward pass variable
$\phi_t(y_t,x)$	: score of node potential number t
$\psi_{t-1}(y_{t-1},y_t,x)$	: score of previous edge potential

2. Backward Pass

Backward Pass almost the same as the forward pass. The only difference is searching from the index t = T to t = 1. Variable Backward denoted $\beta_t[y_t]$.. When t = T, initialize $\beta_T[y_T] = 1/S$ for all labels y_T when t < T, it can be used the following equation [11].

$$\beta_t[y_t] = \mu_t \sum_{y_{t+1}} \beta_{t+1}[y_{t+1}]\phi_{t+1}(y_{t+1},x)\psi_t(y_t,y_{t+1},x) \quad (9)$$

Where :

$\beta_t[y_t]$	: score of backward pass variable number k
μ_t	: scaling factor to ensure $\sum_{y_t}\beta_t[y_t] = 1$
$\beta_{t+1}[y_{t+1}]$	: score of backward variable after
$\phi_{t+1}(y_{t+1},x)$	: score of node potential after
$\psi_t(y_t,y_{t+1},x)$	: score of edge potential number t

3. Node and Edge Probability

To get the probability of the node that calculates the probability of word class label y_t over input data x, can be calculated using the equation [11]:

$$P_t(y_t|x) = w_t\alpha_t[y_t]\,\phi_t(y_t,x)\beta_t[y_t] \quad (10)$$

Where :

$P_t(y_t\|x)$	: node probability number t
w_t	: normalization factor to ensure $\sum_{y_t} P_t(y_t\|x) = 1$
$\alpha_t[y_t]$	: score of forward pass number t
$\phi_t(y_t,x)$	: score of node potential number t
$\beta_t[y_t]$	: score of backward pass number t

Meanwhile, to calculate the edge probability that calculates the probability label y_t for the data x followed by the label y_{t+1} which is a word class label for word thereafter can be calculated using the equation [11]:

$$P_t(y_t,y_{t+1}|x) = \gamma_t\alpha_t[y_t]\,\phi_t(y_t,x)\psi_t(y_t,y_{t+1},x)\phi_{t+1}(y_{t+1},x)\beta_{t+1}[y_{t+1}] \quad (11)$$

$P_t(y_t,y_{t+1}\|x)$	: edge probability number t
γ_t	: normalization factor to ensure $\sum_{y_t} P_t(y_t,y_{t+1}\|x) = 1$
$\alpha_t[y_t]$	: score of forward variable number t
$\phi_t(y_t,x)$	: score of node potential number t
$\psi_t(y_t,y_{t+1},x)$	: score of edge potential number t
$\phi_{t+1}(y_{t+1},x)$	: score of node potential after
$\beta_{t+1}[y_{t+1}]$	: score of backward pass after

1.3.3 Testing

In the testing phase, we want to provide the labels for whole sequence, not just a single node. Theoretically we eant to find the sequence of label that are most likely to happen given data input. In this phase we use method *Viterbi Decoding.* There are two step procedure, that is :

1. Maximal Forward Pass

Maximal Forward Pass : used to compute the probability of the most optimal for data input by considering the relationship between word class label for the words that are being processed by the label word class for the word after. when t = 1, initialize $\boldsymbol{\alpha_1^{max}[y_1]} = \mathbf{1/S}$ for all label $\mathbf{y_1}$ when $\mathbf{t \geq 2}$ we can use the term below.

$$\alpha_t^{max}[y_t] = k_t \max_{y_{t-1}} (\alpha_{t-1}[y_{t-1}^{max}]\phi_t(y_t, x)\psi_{t-1}(y_{t-1}, y_t, x)) \quad (12)$$

Dimana :

$\alpha_t^{max}[y_t]$: score of maximal forward pass $\alpha_t[y_t]$

k_t : scaling factor to ensure $\sum_{y_t} \alpha_t^{max}[y_t] = 1$

$\alpha_{t-1}[y_{t-1}^{max}]$: score od maximal previous forward pass

$\phi_t(y_t, x)$: score of node potential number t

$\psi_{t-1}(y_{t-1}, y_t, x)$: score of previous edge potential

2. Backtracking Pass

The purpose of backtracking is to recover optimal label. The optimal label is computed by the term below.

$$y_t^* = \arg max_{y_t}\left(\alpha_t^{max}[y_t]\phi_t(y_t, x)\right) \quad (13)$$

Sequence $(y_1^*, y_2^*, y_3^*, \dots, y_T^*)$ is the optimal label sequence for input data.

2. RESEARCH CONTENT

2.1 Tagset

Tagset or list part of speech have many list especially for English, such as : 45 for Penn Treebank, 87 for Brown Corpus, and 146 for C7 tagset [2]. As for Indonesian language itself, tagset used less because not all tagset in English can be implemented into Indonesian language. There are several tagset on Indonesian language that is, 25 tagset [12], 19 tagset [13] and 23 tagset [2]. Tagset used in this study was 23 tagset [2], that is :

:

Table 2 Tagset

No.	*Tag*	Description
1	CC	Coordinating conjunction
2	CD	Cardinal number
3	OD	Ordinal number
4	DT	Determiner / article
5	FW	Foreign word
6	IN	Preposition
7	JJ	Adjective
8	MD	Modal and auxiliary verb
9	NEG	negation
10	NN	Noun
11	NNP	Proper noun.
12	NND	Classifier, partitive, and measurement noun
13	PR	Demonstrative pronoun
14	PRP	Personal pronoun
15	RB	Adverb
16	RP	Particle
17	SC	Subordinating conjunction
18	SYM	Symbol
19	UH	Interjection
20	VB	Verbs
21	WH	Question
22	X	Unknown
23	Z	Punctuation

2.2 Data Input

Here is an example of input data used are derived from the corpus [2]:

```
Curah/NN
hujan/NN
yang/SC
tinggi/JJ
menyebabkan/VB
banyak/CD
genangan/NN
,/Z
tempat/NN
ideal/JJ
bagi/IN
nyamuk/NN
aedes/NNP
berkembang biak/VB
./Z

Pemkot/NN
telah/MD
menempatkan/VB
ikan/NN
di/IN
berbagai/CD
genangan/NN
air/NN
untuk/SC
memangsa/VB
telur/NN
nyamuk/NN
./Z
```

Figure 1 Data Input Example

2.3 Preprocessing

Before getting into the main method in the process of POS tagging, done preprocessing beforehand. The preprocessing is carried out as follows.

1. Case Folding

Case folding phase is the conversion of uppercase characters in text to lowercase all. This is done to simplify the search word in the training data.

2. Sentence Tokenization

Sentence tokenization is splitting text into sentences forms with blank lines as separators between sentences.

3. Word Tokenization

Word Tokenization is the separation of the words of the sentence, the separation is done by delimiter spaces, tabs, and new line. While the dot (.), Comma (,) and other punctuation included in word category.

2.4 Process Analyze

Here is a description of a system for POS Tagging that will be built, that is :

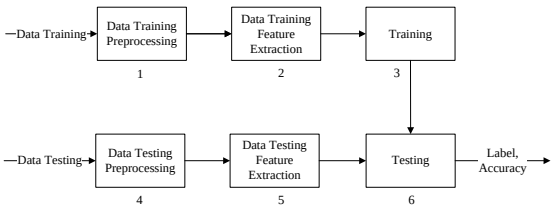


Figure 2 POS *Tagging* Process

3. TESTING

3.1 Parameter Value Testing

At this phase will be tested the influence of parameter values to the resulting accuracy. The value of the parameters tested is the initial value initialization parameter (λ), standard deviation (σ) and the value of learning rate (ω). To initialize the initial value of the value to be tested is 0 and 1, the value of the standard deviation value range is between 2 to 8, while the value of learninng rate between 0.01 to 0.000001. Training data is used contains 200 sentence consisting of a 3910 word derived from the corpus, while testing the data used contains 50 sentences has 908 words.

Here is the result of testing the value of the parameter.

Tabel 3 Hasil Pengujian Nilai Parameter

Initial Value Parameter Feature	Learning rate	Standard Deviation	Accuracy
0	0,01	2	0%
		4	0%
		8	0%
	0,0001	2	60,79%
		4	60,79%
		8	60,79%
	0,000001	2	60,79%
		4	60,79%
		8	60,79%
1	0,01	2	0%
		4	0%
		8	0%
	0,0001	2	77,75 %
		4	77,75 %
		8	77,75 %

3.2 Accuracy Testing

Here is an accuracy test results conducted on four data testing, each composed of 50 sentences from the training of 200 sentences has 3910 words. The value of the parameters used in the training is the value that produces the most excellent accuracy of the testing parameter values process. For the initial initialization parameter features values is 1, standard deviation value is 2 and the value of learning rate is 0.000001. As for the counting accuracy for each of the data testing is as follows.

$$\text{Accuracy} = \frac{\text{number of true tag prediciton}}{\text{Number of words in data testing}} \text{ x } 100\%$$

The following is the calculation accuracy for Data Testing 1.

$$\text{Accuracy} = \frac{\text{number of true tag prediciton}}{\text{Number of words in data testing}} \text{ x } 100\%$$
$$= \frac{706}{908}\, x\, 100\ \%$$
$$= 77{,}75\ \%$$

As for the test results to the entire data testing can be seen in the following table.

Table 4 Accuracy Testing Result

Data Testing	Number of Words	Number of True Tag Prediciton	Accuracy
1	908	978	77,75%
2	949	732	77,13 %
3	1009	793	78,59%
4	2902	1467	79,31%
Rata-rata			78,20%

4. CONCLUSION AND ADVICE

Based on the results of tests performed on this research, the average accuracy generated by the POS tagger to Indonesian by using CRF is equal to 78.20%.

The advice in the development of POS Tagger for Indonesian by using CRF are as follows.

1. The addition of a node feature that can increase the probability of a word given a specific label.
2. Implementation of other training methods, for knowing which training method is more accurate to use the POS tagger Indonesian by using CRF.
3. Optimization program that can process more training data.

5. REFERENCES

[1] Jurafsky, D, Martin, J.H. Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. New Jersey : Prentice-Hall Inc. 2000.

[2] Dinakaramani, A, Et. al. *Designing an Indonesian Part of Speech Tagset and Manually Tagged Indonesian Corpus*. 2014.

[3] Pisceldo, F, Adriani, M, Manurung, R. *Probabilistic Part of Speech Tagging for Bahasa Indonesia*. 2009.

[4] Wicaksono, A.F., Purwanti, A. HMM Based POS Tagger for Bahasa Indonesia, Fourth International MALINDO Workshop.

[5] Chandrawati, T. Pengembangan Part of Speech Tagger untuk Bahasa Indonesia Berdasarkan Metode Conditional Random Fields dan Transformation Based Learning. Universitas Indonesia. 2008.

[6] Rashel, F, Luthfi, A, Dinakaramani, A, Manurung, R. Building an Indonesian Rule-Based Part-of-Speech Tagger. 2014.

[7] Manurung, R. Tutorial: Pengenalan terhadap POS Tagging dan probabilistic Parsing. Workshop Nasional INACL, Jakarta. 2016.

[8] Lafferty, J, McCallum, A, Pereira, F. Conditional Random Field: Probabilistic Model for Segmenting and Labeling Sequence Data. 2001.

[9] Wallach, M. Hanna. Conditional Random Fields : An Introduction. 2004.

[10] Klinger, R, Tomanek, K. Classical Probabilistic Model and Conditional Random Field. 2007.

[11] Truyen, T.T, Phung, D. A Practitioner Guide to Conditional Random Field for Sequential Labeling. 2008.

[12] M. Adriani, R. Manurung, and F. Pisceldo, "Statistical Based Part Of Speech Tagger for Bahasa Indonesia", in Proceedings of the 3rd International MALINDO Workshop, Co-located Event ACL-IJCNLP 2009. Singapore, August 1,2009.

[13] S. D. Larasati, V. Kuboň, and D. Zeman, "Indonesian Morphology Tool (MorphInd): Towards an Indonesian Corpus", in Proceedings of the Workshop on Systems and Frameworks for Computational Morphology (SFCM 2011). August 2011. Zurich, Switzerland

[14] Kridalaksana, H. Kelas Kata Dalam Bahasa Indonesia Edisi 2. Jakarta : Gramedia. 2008.