A Performance Evaluation of Classifiers Employ Language Dependent Tools for Indonesian Text

Surya Sumpeno, *Student Member, IEEE,* Agus Zainal Arifin, *Member, IEEE* Mochamad Hariadi, *Member, IEEE* and Mauridhi Hery Purnomo, *Member, IEEE*

***Abstract*—This paper evaluates the performance of Maximum Entropy (MaxEnt), Support Vector Machine (SVM) and Naïve Bayes (NB) techniques for Indonesian text classification. Performance of MaxEnt and SVM techniques are compared against baseline NB technique. We also investigate the effect of language dependent tools such as Indonesian stemming and stop words removal can have on these techniques for text classification performances. Up to now, there is no experimental report about the effect of Indonesian stemmer on the text classification accuracy. From our experiments, we conclude that maximum entropy performs better than other classifiers in general. Language dependent tools such as stemming and stop words removal have only little effect on the accuracy of text classification. However stemmed approach scored highest average accuracy and due to the dimension reduction of feature vectors used in classification, make this approach is viable step in pre-processing stage.**

***Index Terms*—maximum entropy, support vector machine, naïve bayes, indonesian text classification, language dependent tool, stopwords removal, stemming**

I. Introduction

Text mining is of growing importance as the volume of unstructured text in web pages, digital libraries and community wide intranets continue to increase. Text clustering and automatic text classification are considered as important applications in text mining.

Text clustering aims to discover natural groupings, and thus present an overview of the classes (topics) in a collection of documents. In the field of artificial intelligence, this is known as unsupervised machine learning. While automatic text classification is a process where the number of classes (and their properties) are known a priori and documents are then assigned to these classes [1]. Automatic text classification have been used in many applications such as e-mail filtering [2], topic identifications [3], automatic meta-data organization, text filtering and documents' organization for databases and web pages [4].

The dominant approach to automatic text classification is based on machine learning techniques: a general inductive process automatically builds a classifier by learning, from a set of preclassified documents, the characteristics of the categories. The advantages of machine learning approach over the knowledge engineering approach (consisting in the manual definition of a classifier by domain experts) are a very good effectiveness, considerable savings in terms of expert manpower, and straightforward portability to different domains [5].

Many techniques for supervised learning algorithms for text classification have showed reasonable performance. These techniques include Naïve Bayes [6], k-nearest neighbor (k-nn) [7], Support Vector Machines (SVM) [8], boosting [9] and rule learning algorithms [10]. Among these, however, no single technique has proven to consistently outperform the others across many domains [11].

This paper evaluates the performance of Maximum Entropy (MaxEnt) and SVM techniques for Indonesian text classification and compare their performance against baseline Naïve Bayes technique. We also investigate the effect of language dependent tools such as Indonesian stemming and stop words removal can have on both techniques for text classification performances.

II. Indonesian Stemming and StopWords Removal Effect on the Performance of Text Classification

It is generally believed that applications such as information retrieval, text classification, or document filtering could benefit from the existence and availability of basic tools of Natural Language Processing (NLP) such as stemmers, morphological analyzers or part-of-speech taggers [12]. Stemming is clearly language dependent.

The most important property of a stemmer for text classification is the number of words it correctly reduces to the same stem. Stemming reduces all inflected forms of a word to the same stem. Based on the assumption that terms which have a common stem will usually have similar meaning, the stemming process is used in text classification as one possible way to improve performance. Just as in text classification, stemming is also used in Information Retrieval (IR) system where the main goal is to retrieve the documents that correspond to a given query. However, stemming for IR system has been reported to show mixed results.

Performance comparison has been conducted on data stemmed with three suffix-stripping algorithms for English against unstemmed data in IR queries and [13] concluded that stemming does not consistently improve performance. Derivative stemming in Spanish [14] caused a worse performance of IR than no stemming.

On the other hand, [15] and [16], both concluded that stemming of English does improve performance of an IR system. Investigation have been conducted [17] to check whether

Surya Sumpeno, Mochamad Hariadi and Mauridhi Hery Purnomo are with the Electrical Engineering Department, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, 60111 e-mail:(see http://blog.its.ac.id/surya).

Agus Zainal Arifin is with the Faculty of Information Technology, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, 60111 e-mail:(see http://agusza.if.its.ac.id/).

Manuscript received 2010; revised 2010.

stemming would have more effect for a morphologically complex language like Slovene. They found that precision of the retrieved documents was increased when suffix-stripping was used.

In the case of Bahasa Indonesia, there have been some implementations of Indonesian stemmer: [18], [19], [20], and [21]. Using her own Indonesian stemming algorithm, [22] reported Indonesian stemming for text retrieval has little effect on accuracy. In text clustering, due to the requirement of online processing [21] only interested in faster processing time as an effect of terms reduction by stemming. Using k-means clustering, [23] concluded the best performance is achieved without stemming.

Up to now, there is no experimental report about the effect of Indonesian stemmer on the text classification accuracy. Our motivation is to investigate the stemming and stop words removal effect on Indonesian text classification tasks. We employ Indonesian stemming developed by [21] to stem Indonesian words in the documents for our experiments and make use of the list of defined Indonesian stop words in the paper.

III. Maximum Entropy for Text Classification

Maximum entropy has already been widely used for a variety of NLP tasks, such as prepositional phrase attachment [24], part-of-speech tagging [25], language modeling [26] and text segmentation [27]. Maximum entropy has been shown to be a practical and competitive algorithm in these domains.

The maximum entropy model estimates probabilities based on the principle of making as few assumptions as possible, other than the constrained imposed. The principle in maximum entropy is that when nothing is known, the distribution should be as uniform as possible, hence maximum entropy will be achieved [28].

Labeled training data is used to determine a set of constraints for the model that characterize the class-specific expectations for the distribution. Constraints are represented as expected values of "features," any real-valued function of an example. In our text classification experiment, a document is represented by a set of word count features. Maximum entropy then estimates the conditional distribution of the class label given a document.

Maximum entropy is a supervised learning technique. Therefore, it needs a training corpus (labeled data). The labeled training data is used to set constraints on the conditional distribution and estimate the expected value of these word counts on a class-by-class basis. Improved iterative scaling will find a text classifier of an exponential form that is consistent with the constraints from the labeled data.

Each constraint expresses a characteristic of the training data that should also be present in the learned distribution. In text classification, maximum entropy is a model which assigns a class c of each word w based on its document d in the training data D. The learned conditional distributed $P(c|d)$ is computed as follows:

$$P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \alpha_i f_i(d,c)), \tag{1}$$

where each $f_i(d,c)$ is a feature, α_i is a parameter to be estimated and $Z(d)$ in equation (1) is a normalization function which is computed as follows:

$$Z(d) = \sum_c \exp(\sum_i \alpha_i f_i(d,c)). \tag{2}$$

The parameter α_i in equation (2) can be estimated by an iterative way using Improved Iterative Scaling (IIS), a hillclimbing algorithm for calculating the parameters of a maximum entropy classifier given a set of constraints. A more complete explanation of improved iterative is presented by [29].

Given a set of training data D, the log likelihood Λ of the exponential model (1) is

$$\begin{aligned} l(\Lambda|D) \quad = \quad & \sum_{d\in D}\sum_i \alpha_i f_i(d,c(d)) - \\ & \sum_{d\in D} \log \sum_c \exp \sum_i \alpha_i f_i(d,c). \end{aligned} \tag{3}$$

IIS algorithm starts from any initial vector of parameters Λ. At each step Λ is improved by setting it equal to $\Lambda+\Delta$, which will have a higher likelihood. Thus, at each step, we want to find a Δ such that the difference in likelihoods is positive:

$$l(\Lambda+\Delta|D) - l(\Lambda|D) > 0. \tag{4}$$

Using the inequality $-\log(x) \geq 1 - x$ and Jensen's inequality, the expression (5) is bounded with an auxiliary function called B:

$$\begin{aligned} l(\Lambda \quad + \quad & \Delta|D) - l(\Lambda|D) \geq B = \\ & 1 + \sum_{d\in D} (\sum_i \delta_i f_i(d,c(d)) - \\ & \sum_c P_\Lambda(c|d) \exp(f^\sharp(d,c)\delta_i \sum_i \frac{f_i(d,c)}{f^\sharp(d,c)})). \end{aligned} \tag{5}$$

The best Δ can be found by diferentiating B with respect to the change in each parameter δ_i in turn and get the maxima:

$$\begin{aligned} \frac{\partial B}{\partial \delta_i} \quad = \quad & \sum_{d\in D} (f_i(d,c(d)) - \\ & \sum_c P_\Lambda(c|d) f_i(d,c) \exp(\delta_i f^\sharp(d,c))). \end{aligned} \tag{6}$$

In our experiments, word counts are used as features. More precisely the feature is formulated as:

$$f_{w,c'}(d,c) = \begin{cases} 0 & \text{if } c \neq c' \\ \frac{N(d,w)}{N(d)} & \text{Otherwise,} \end{cases} \tag{7}$$

where $N(d,w)$ is the number of times word w occurs in document d, and $N(d)$ is the number of words in d.

With a limited number document collections, the expected value of a feature in the training data may be far from the real value. By introducing a prior on the maximum entropy model, overfitting can be reduced and hence performance will be improved.

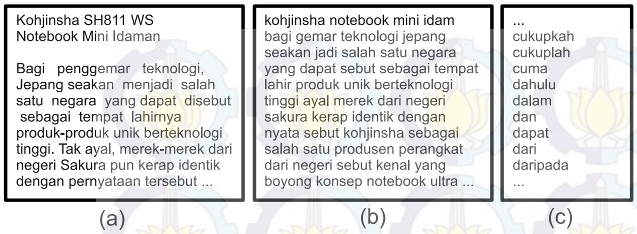

Fig. 1. (a) Sample of text "as-is" (b) Sample of stemmed text (c) Part of stop words.

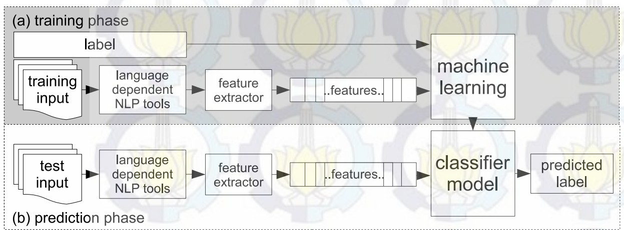

Fig. 2. Supervised Classification. Input is (optionally) first preprocessed using language-dependent tool, such as stemming and stop words removal. (a) During training, a feature extractor is used to convert each input value to a feature set. Pairs of feature sets and labels are fed into the machine learning algorithm to generate a model. (b) During prediction, the same feature extractor is used to convert unseen inputs to feature sets. These feature sets are then fed into the model, which generates predicted labels. During test, these predicted labels will be matched against ground-truth labels.

The prior probability of our model is the product over the Gaussian of each feature value α_i with variance σ_i^2 then added as single term to Equation (6):

$$\frac{\partial B}{\partial \delta_i} = \frac{\alpha_i + \delta_i}{-\sigma_i^2} + \sum_{d \in D} (f_i(d, c(d)) - \sum_c P_\Lambda(c|d) f_i(d,c) \exp(\delta_i f^\sharp(d,c))). \quad (8)$$

IV. Support Vector Machine for Text Classification

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression, which adhere to Vapnik's structural risk minimization principle (SRM). SRM is an inductive principle for model selection used for learning from finite training data sets, which describes a general model of capacity control and provides a trade-off between hypothesis space complexity and the quality of fitting the training data (empirical error).

SVM generalization (the ability of a hypothesis to correctly classify data not in the training set) performance, namely error rates on test sets, either matches or is significantly better than that of competing classification methods [30]. That is why we included SVM in our experiments.

At its heart, SVM defines many hyperplanes (linear classifiers) that separate the data, as shown in Figure 3. However only one of these achieves maximum separation, in other words, we seek maximum margin classifier or hyperplane as an apparent solution. Suppose we have some hyperplane H which separates the positive from the negative examples (a separating hyperplane). The points x which lie on the hyperplane satisfy $w \cdot x + b = 0$, where w is normal to the hyperplane, $|b|/\|w\|$ is the perpendicular distance from the hyperplane to the origin, and $\|w\|$ is the Euclidean norm of w.

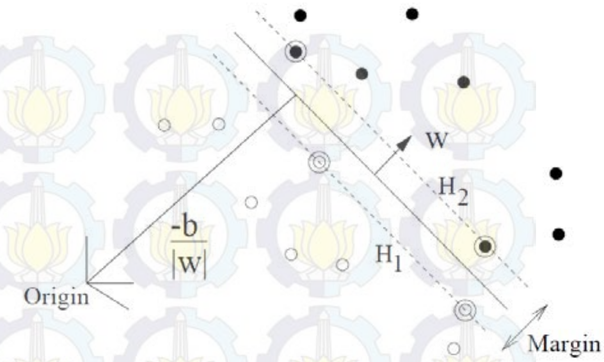

Fig. 3. Linear separating hyperplanes for the separable case. The support vectors are circled.

For nonlinear data, using the right kernel K mapping technique, we will arrive at solution.

The first step in text classification is to transform text documents, which typically are strings of characters, into a representation suitable for the machine learning algorithm and the classification task.

Each distinct word corresponds to a feature x, with the number of times word occurs in the document as its value. To avoid unnecessarily large feature vectors, words are considered as features only if they occur in the training data at least 3 times and if they are not "stop-words" (like "dan", "atau", etc.). Stemming can be employed in pre-processing step to further reduce the number of feature vectors.

In our experiments, we used C-SVC and Radial Basis Function (RBF) kernel, implemented in RapidMiner [1] which based on [31] for multiclass classification. A term frequency-inverse document frequency (TF-IDF) technique is utilized to transform documents into feature vectors. We have conducted extensive k-fold cross-validations and found $\gamma = 1$ is the parameter to achieve the best accuracy.

V. Naïve Bayes for Text Classification

In our experiments, Naïve Bayes (NB) is used as a baseline, because it is fast, easy to implement and eventough independence is generally a poor assumption, in practice NB often competes well with more sophisticated classifiers. NB is also categorized as a supervised technique as well as MaxEnt and SVM.

A widely used NB for classification is provided by a simple theorem of probability known as *Bayes' rule* which according to [6] can be simplified as:

$$P(c_k|x) = P(c_k)\frac{P(x|c_k)}{P(x)}. \quad (9)$$

When $P(c_k|x)$ is known exactly for a classification problem, classification can be done in an optimal way, for instance

[1] http://rapid-i.com/

the expected number of classification errors can be minimized by assigning a document with feature vector x to the class c_k for which $P(c_k|x)$ is highest. Most often, $P(c_k|x)$ is not known and must be estimated from data, which is difficult to be done directly. A common way to overcome this difficulty is to assume that the distribution of x conditional on c_k can be decomposed for all c_k and written as:

$$P(x|c_k) = \prod_{j=1}^{d} P(x_j|c_k). \tag{10}$$

If we assume the independence of occurence (that is why this particular method is called Naïve), that is the occurence of a particular value of x_j is statistically independent of the occurence of any other $x_{j'}$ given a document of type c_k, then Equation 9 becomes:

$$P(c_k|x) = P(c_k)\frac{\prod_{j=1}^{d} P(x_j|c_k)}{P(x)}. \tag{11}$$

If all values of the right side are estimated (indicated by hats) then we have an estimate for $P(c_k|x)$:

$$\widehat{P(c_k|x)} = \frac{\widehat{P(c_k)}\prod_{j=1}^{d}\widehat{P(x_j|c_k)}}{\widehat{P(x)}}. \tag{12}$$

If the goal of classification is to minimize number of errors, then a document with feature vector x can be assigned to the each c_k such that $\widehat{P(c_k|x)}$ is highest. This is a NB classifier. Typically denominator $\widehat{P(x)}$ can be omitted from computation, because it is the same for all c_k. In practice, classification will still be considered accurate as long as the correct class has the highest value of numerator $\widehat{P(c_k)}\prod_{j=1}^{d}\widehat{P(x_j|c_k)}$.

In our experiments, we employ *Bag-of-Words* (BoW) model, feature x_j is associated with w unique words observed in the collection of documents. Then the full representation of document is $X = (x_1, ..., x_j, ..., x_w)$, where all x_j is integer word count. We use the multinomial instantiation of NB implementation [33] as experiment tool.

For Indonesian text classification, [32] utilized NB to clasifiy Indonesian news to six classes. The best result has been reported with 90.23% accuracy using 407 documents as training samples and 175 documents as test (the ratio of training to test is about 70:30). However stemming effect has not yet been investigated.

VI. Experiments and Discussions

In our system, we (optionally) first preprocessed data using dependent-language techniques such as tokenizing, stemming and stop words removal, before feed it into machine learning algorithm. The structure of the system is depicted in Fig. 2.

Our data set contains full-text articles from CHIP[2], monthly Indonesian computer magazine. Originally they came in PDF format and we manually extract only the text. These document collections then are divided manually into four classes: `perangkat-keras (hardware)`, `perangkat-lunak (software)`, `berita (news)` and `tips`, all together containing 4,320 documents, with each class consist of 1,080 text documents.

[2] http://www.chip.co.id/

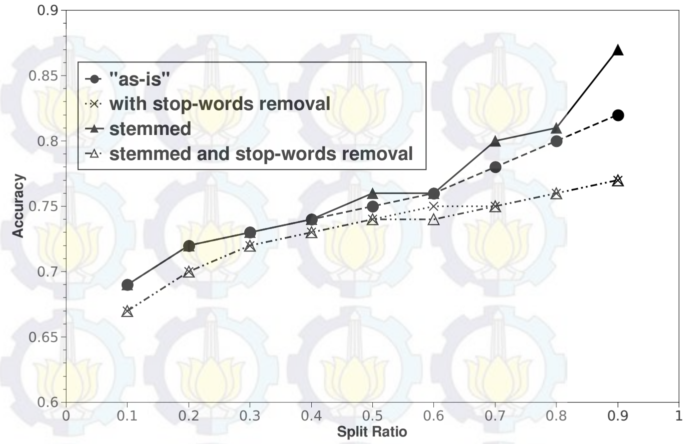

Fig. 4. Performance of Maximum Entropy

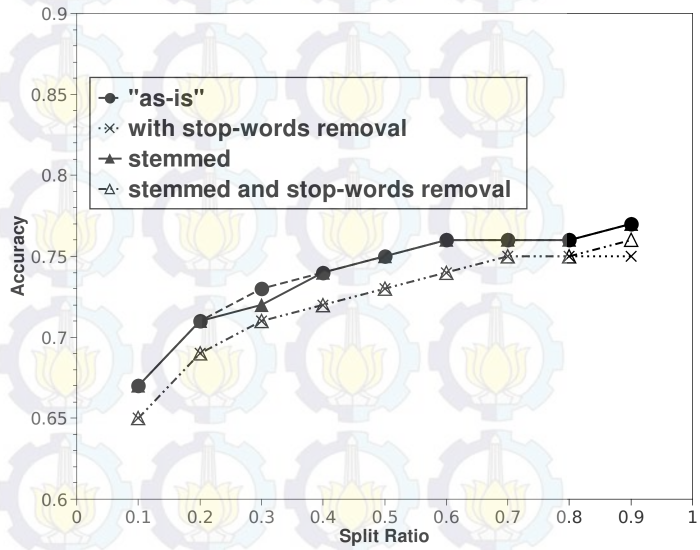

Fig. 5. Performance of SVM

We run 10-folds cross validation for MaxEnt, SVM and NB techniques, i.e. divide data set into 10 sub-samples. The ratio of training to test begins at 10%, raising 10% until reaches 90%. For each split ratio, we conduct 10 trials to get the average score of accuracy.

There are 3 types of input data: text "as-is" (with no language-dependent pre-processing at all), text with stop words removal, and stemmed text with stop words removal. These 3 types of input data are included in the experiments to investigate the effect of language dependent tools, namely stemming and stop words removal, on text classification tasks using MaxEnt, SVM dan NB. Fig. 1 (a) and (b) depicts sample of document text "as-is" and stemmed text respectively, and Fig. 1 (c) depicts part of our stop words used in our experiments. We compile stop words by hand, carefully include word which by human judgement does not affect classification, such as preposition words for example *"yang"* ("which", "that"),

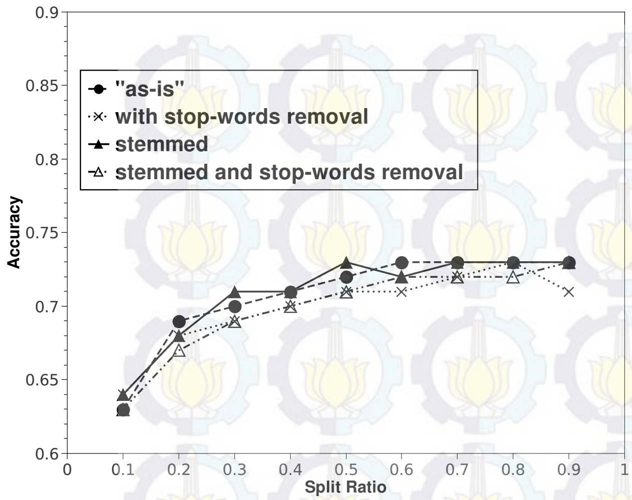

Fig. 6. Performance of NB

"tentang" ("about"), and other words for example *"cukup"* ("enough"), *"juga"* ("also"). Our stop words list include 793 words.

For MaxEnt, prior is set to 0.1. Fig. 4 depicts performance of MaxEnt for text classification, while Fig. 5 and Fig. 6 depicts performance of SVM and NB respectively. As shown, performance tends to get better when number of training samples raises. For method, MaxEnt performs better than all other classifiers. Average accuracy of MaxEnt, SVM and NB are equal to 75%, 73% and 70% respectively.

It is interesting to see that using stop words removal and stemming in combination, tend to degrade the performances of all classifiers by a small fraction, while performances of no pre-processing at all ("as-is") generally equal to stemmed approach's, with the exception of MaxEnt which has highest average accuracy (equals to 76%). Average accuracy for stemmed approach of MaxEnt, SVM and NB are equal to 75%, 73% and 70% respectively. Please note that stemming reduces the number of feature vectors used in classification. Stopwords removal approach generally degrades performances of text classifiers and shows the worst accuracies.

VII. Conclusion and Future Work

This paper has presented a performance evaluation of text classifiers, namely MaxEnt, SVM and NB for Indonesian text. The use of language dependent tools such as stop-words removal and stemming affect only a little to their performances. However, stemmed approach always shows best performances and its desirable characteristics, such as feature vectors dimension reduction, will make this approach preferred approach in pre-processing step of Indonesian text classification.

Our experimental results show that MaxEnt is a technique that warrants further investigation for text classification. For future work, we will utilize MaxEnt to other tasks in Natural Language Processing (NLP) fields such as classifying emotional response from text input and evaluate its performance.

References

[1] N.O. Andrews and E.A. Fox, "Recent developments in document clustering.," Technical Report TR-07-35, 2007.
[2] W.W. Cohen, "Learning rules that classify e-mail," In Papers from the AAAI Spring Symposium on Machine Learning in Information Access, pp.18–25, AAAI Press, 1996.
[3] S. Tiun, R. Abdullah, and T.E. Kong, "Automatic topic identification using ontology hierarchy," In the Proc. of the 2 nd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2001), pp.444–453, 2001.
[4] Y. Yang, S. Slattery, and R. Ghani, "A study of approaches to hypertext categorization," Journal of Intelligent Information Systems, vol.18, pp.219–241, 2002.
[5] F. Sebastiani and C.N.D. Ricerche, "Machine learning in automated text categorization," ACM Computing Surveys, vol.34, pp.1–47, 2002.
[6] D.D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," pp.4–15, Springer Verlag, 1998.
[7] Y. Yang, "An evaluation of statistical approaches to text categorization," Journal of Information Retrieval, vol.1, pp.67–88, 1999.
[8] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," pp.137–142, Springer Verlag, 1998.
[9] R.E. Schapire and Y. Singer, "Boostexter: a boosting-based system for text categorization," Machine Learning, pp.135–168, 2000.
[10] W.W. Cohen and Y. Singer, "Context-sensitive learning methods for text categorization," ACM Transactions on Information Systems, pp.307–315, ACM Press, 1996.
[11] K. Nigam, "Using maximum entropy for text classification," In IJCAI-99 Workshop on Machine Learning for Information Filtering, pp.61–67, 1999.
[12] L. Asker, A.A. Argaw, M. Sahlgren, and B. Gambäck, "Applying machine learning to amharic text classification," Proceedings of WOCAL 5: 5th World Congress of African Linguistics, Addis Ababa University, Ethiopia, 2006.
[13] D. Harman, "How effective is suffixing?," Journal of the American Society of Information Science, vol.42, no.1, pp.7–15, 1991.
[14] C.G. Figuerola, R. Gómez Díaz, Á.F. Zazo Rodríguez, and J.L. Alonso Berrocal, "Stemming in Spanish: A first approach to its impact on information retrieval," Results of the CLEF 2001 Cross-Language System Evaluation Campaign. Working Notes for the CLEF 2001 Workshop. 3 September, Darmstadt, Germany, pp.197–202, 2001. [También en línea: http://www.ercim.org/publication/ws-proceedings/CLEF2/figuerola.pdf - Consulta: 12.01.2003].
[15] R. Krovetz, "Viewing morphology as an inference process," SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, pp.191–202, ACM, 1993.
[16] D. Hull, "Using statistical testing in the evaluation of retrieval experiments," SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, pp.329–338, ACM, 1993.
[17] M. Popovic and P. Willett, "The effectiveness of stemming for natural-language access to slovene textual data," JASIS, vol.43, no.5, pp.384–390, 1992.
[18] J. Asian, H.E. Williams, and S.M.M. Tahaghoghi, "Stemming indonesian," ACSC '05: Proceedings of the Twenty-eighth Australasian conference on Computer Science, Darlinghurst, Australia, Australia, pp.307–314, Australian Computer Society, Inc., 2005.
[19] F.Z. Tala, "A study of stemming effects on information retrieval in bahasa indonesia," M.S. thesis, 2003.
[20] V.B. Vega and S. Bressan, "Stemming indonesian words without a dictionary," The Eleventh International Symposium On Malay/Indonesian Linguistics (ISMIL 11), 2007.
[21] A.Z. Arifin, R. Darwanto, D.A. Navastara, and H.T. Ciptaningtyas, "Online indonesian news classification using suffix tree clustering algoritm," Seminar Sistem Informasi Indonesia, Institut Teknologi Sepuluh November, Surabaya, Indonesia, Information System Department, December 2008.
[22] J. Asian, H.E. Williams, and S.M.M. Tahaghoghi, "A testbed for indonesian text retrieval," ADCS, ed. P. Bruza, A. Moffat, and A. Turpin, pp.55–58, University of Melbourne, Department of Computer Science, 2004.
[23] Y. Wibisono and M.L. Khodra, "Indonesian news clustering," KNSI Konferensi Nasional Sistem Informasi, Bandung, Indonesia, UNPAS Pasundan University, February 2006.

[24] A. Ratnaparkhi, J. Reynar, and S. Roukos, “A maximum entropy model for prepositional phrase attachment,” In Proceedings of the ARPA Workshop on Human Language Technology, pp.250–255, 1994.
[25] 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.
[26] S. Chen and R. Rosenfeld, “A gaussian prior for smoothing maximum entropy models,” 1999.
[27] D. Beeferman, A. Berger, and J. Lafferty, “Statistical models for text segmentation,” Mach. Learn., vol.34, no.1-3, pp.177–210, 1999.
[28] A. Berger, S. Pietra, and V. Pietra, “A maximum entropy approach to natural language processing,” 1996.
[29] S.D. Pietra, V.D. Pietra, and J. Lafferty, “Inducing features of random fields,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.19, pp.380–393, 1997.
[30] C.J.C. Burges, “A tutorial on support vector machines for pattern recognition,” Data Min. Knowl. Discov., vol.2, no.2, pp.121–167, 1998.
[31] C.C. Chang and C.J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[32] Y. Wibisono, “Indonesian news classification using naïve bayes classifier,” Seminar Nasional Matematika UPI, Bandung, Indonesia, UPI Universitas Pendidikan Indonesia, 2005.
[33] A. Mccallum, “A comparison of event models for naive bayes text classification,” In AAAI-98 Workshop on Learning for Text Categorization, pp.41–48, AAAI Press, 1998.