khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Automatic Language Identification for Indonesian-Malaysian Language Using Machine Learning

Abdiansah Abdiansah[1*], Muhammad Qurhanul Rizqie[2]
Departement of Computer Science
Universitas Sriwijaya
Palembang, Indonesia
*abdiansah@unsri.ac.id

Abstract-Language Identification (LID) aims to guess or identify which language the text or sound is coming from. Language identification tends to be easier in languages with different characteristics (e.g., Indonesian and English), but not for languages with similar characteristics (e.g., Indonesian and Malaysian). Similar languages can cause ambiguity that will be a bias for machine learning. Using Support Vector Machine (SVM) technique, this research tried to identify the Indonesian or Malaysian language. The training and testing data are taken from Leipzig Corpora Collection and Twitter dataset. The feature representation technique uses TF-IDF, and the baseline testing uses Naive Bayes Multinomial. We used two training techniques: split (20:80) and 10-cross validation. The experimental results show that the accuracy between the baseline and SVM is not too far. Both provide accuracy of around 90% and above. The results indicate that Indonesian and Malaysian language identification accuracy is relatively high even though using simple techniques.

Keywords: Language Identification, Indonesian, Malaysian, Support Vector Machine

Article info: *submitted January 31, 2023, revised August 6, 2023, accepted August 7, 2023*

1. Introduction

Language Identification (LID) is a topic in Natural Language Processing (NLP) that aims to automatically recognize the language used, whether it comes from sound, document, or writing. Although scientists have researched this topic, this topic has become active again since the emergence of the social media era [1]. The current language identification challenge is short, non-standard text and contains a lot of noise (misspellings, non-word tokens such as URLs, emoticons, hashtags, and foreign language words). Examples include text on Twitter, captions on Instagram, and posts/comments on Facebook. Previously, many language identification studies used long and noise-free text. Unfortunately, the significant accuracy decreased when the technique was applied to the short texts from social media [2].

Language identification plays a vital role in natural language-based systems with inputting more than one language (at least two languages). Usually, language identification is put at the beginning of processing (before pre-processing) to determine the right strategy for the following process. The following are examples of applications that apply language identification: filter documents by language (e.g., Google Playbook), automatic language detection on machine translators (e.g., Google Translate), language detection based on voice (e.g., Google Assistant); filter comments/reviews, (e.g., Goodreads), and separate data in datasets originated from an automatic retrieval system (e.g., the Twitter dataset [3] containing Indonesian and Malaysian text).

Automatic language identification of two languages with different degrees will be easier because the distinguishing characteristics are pretty good. The difference between the two languages lies in the pronunciation at the word level. The problem arises when the two languages have similar characteristics that give rise to linguistic phenomena, such as the similar word, ambiguous word, and others. The two languages are closely related because they have the same characteristics at the word level. These linguistic phenomena can be a problem for machine learning during training and identification.

Research on language identification for Indonesia-Malaysia is quite rare. We have conducted a literature study and only obtained three studies focusing on this area. The first study was started in 2006 by Ranaivo-Malancon [4]. He used a statistical approach to identify Indonesian or Malaysian words using four variables: word frequency, trigrams, exclusive words, and number formats.

Furthermore, in 2015, Indra et al. [5] investigated a similar topic using a self-developed language identification algorithm with a pipeline processing approach. The algorithm does not carry out the training process. Finally, in 2018, Nomoto et al. [6] classified Indonesian and Malaysian languages based on the Leipzig corpus collection using a simple decision tree. So far, prior studies show that they only used simple techniques to solve the problem. In this study, we used a machine learning approach.

Research in the field of natural language processing using machine learning has been studied by several researchers, from general models [7], [8] to more complex models (deep learning) [9]. The domains that apply machine learning methods to natural languages are also very diverse [10]–[12]. In general, research on language identification still focuses on the challenges of the social media corpus, such as Twitter [2][13][14] and the use of deep learning methods [15]–[17].

Based on the description above, the following explains why research on identifying Indonesian-Malaysian languages is essential: (1) Indonesian and Malaysian languages have a close relationship, so many words are similar. It can be a bias for machine learning algorithms and a challenge for researchers to eliminate the bias; (2) The training text datasets generated automatically from the Internet usually contain a mixture of Indonesian and Malaysian, and even other foreign languages such as Arabic and Chinese. It can reduce the performance of machine learning specific to the Indonesian language. (3) So far, there is no research about Indonesian-Malaysian language identification using the machine learning approach. Therefore, this study's results can be used as a benchmark for further research. Our primary contribution for the field of Indonesia Malaysian Automatic Language Identification is open public dataset and baseline result using Naïve Bayes and Support Vector Machines.

There are two works carried out in this research: (1) Building a dataset for the identification of Indonesian-Malaysian languages, especially for machine learning training data; and (2) Developing and testing an Indonesian-Malaysian language identification system using machine learning. Support Vector Machines (SVM) is used as a machine learning method because it is suitable for classifying two classes. We also use the Naive Bayes algorithm as the baseline model. Naive Bayes is a straightforward machine learning method that is often compared with other methods.

2. Methods

a. Dataset

This study uses primary and secondary datasets. The training dataset uses secondary dataset taken from Leipzig Corpora Collection corpus [18]. We downloaded two text files from the web: (1) ind-id_web_2013_1M-sent is for the Indonesian language dataset; and (2) msa-my_web_2013_1M-sent is for the Malaysian language dataset. Each dataset contains one million sentences. Furthermore, the test dataset uses primary dataset taken from the Goodreads website for long texts and the dataset from ferdiana et al. [3] for short texts. Ferdiana et al. built a dataset using automatic crawler tools. Based on our analysis, we found that the dataset contains Indonesian and Malaysian languages. So that we manually separated the data to meet our experimental requirement. After that we add data from Goodreads to combined with Ferdiana's dataset. Figure 1 shows the corpus of both train and test dataset.

b. Building Dataset

This training dataset in this study is taken from secondary data. However, several manual processes are done to make the data suitable for SVM's features, for examples removing serial numbers and eliminating types of words other than Latin (Arabic, Chinese, etc.) and others. The test dataset is taken from primary data by collecting data manually from the Internet and then performing pre-processing to remove text noises. Finally, the labeling process determines whether the text is Indonesian or Malaysian.

Leipzig Corpora Collection (LCC) is a collection of texts in the form of sentences separated by lines. In addition, there are still lots of annoying noise, such as foreign characters (Arabic, Chinese, Japanese, etc.), numbers, and unusual characters. Therefore, the corpus must be cleaned and created in a standard dataset format containing attribute names and data separators using commas (CSV).

There are six steps to building an Indonesian-Malaysian language identification dataset: (1) Get 10000 data from Indonesian and Malaysian corpus, for example, ind-5K.txt and msa-5K.txt; (2) Remove all characters other than letters and spaces using regex [^ a-zA-Z] +, remove excess spaces, and convert to CSV file; (3) Add ID and CATEGORY (use a spreadsheet) and fill the category column with INDONESIA or MALAYSIA label, according to their respective corpus; (4) Randomized the ID for Indonesian and Malaysian datasets, then combined the Indonesian and Malaysian datasets; (5) Sort the ID and randomize the dataset. It makes the ID unique; and (6) Finally, save the dataset as ds-ind-msa-10K.csv.

The ds-ind-msa-10K.csv dataset contains 10.000 lines of text consisting of 5000 data labeled INDONESIA and 5.000 data labeled MALAYSIA. In contrast, The Leipzig Corpora Collection corpus contains one million lines of text. Therefore, it is possible to create a dataset of up to two million lines of text, a combination of Indonesian and Malaysian corpora. Our work only used 10.000 sentences because limitation of labeling cost.

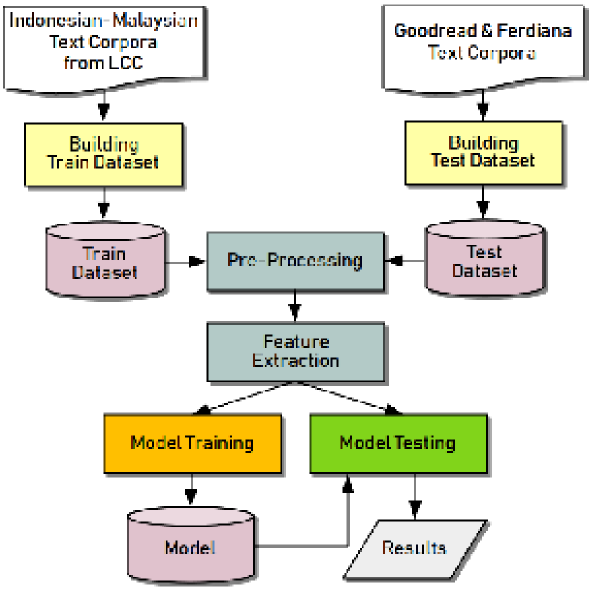


Figure 1. Proposed model of ind-my language identification

c. Proposed Model

We used Support Vector Machine (SVM) [19], [20] and Naive Bayes Multinomial [21] as machine learning methods to identify whether a text belongs to the Indonesian or Malaysian language. In general, the stages of our works in automatic language identification using machine learning are shown in Figure 1. There are four main steps: (1) Building train and test datasets from corpora; (2) Doing pre-processing and features extraction for machine learning; (3) Training the machine learning model (SVM and Naive Bayes); and (4) Testing the model that yields from the previous step.

The first step is to build a train and test dataset from corpora. As we said before, we used Indonesian-Malaysian corpora from LLC for a training dataset and from Goodread's web and Ferdiana dataset for a testing dataset. After we have both datasets, the next step is pre-processing, such as normalization, tokenization, noise removal, etc. The output of this process is text that is considered to be free from noise. The next step is to perform feature extraction so that it is suitable to be a machine learning input. We use the TF-IDF method to represent text as vectors. The output of this process is a collection of vectors. The last step is training the machine learning using a training dataset to produce a model. The last step is testing the model. The output of model testing is language identification that results from SVM or Naive Bayes.

We use the Naive Bayes Multinomial method for baseline. Naive Bayes is famous and simple for text classification problems. Also, it is fast computation, and sometimes the results are better than complex methods. We use the split approach (80:20) and 10-cross validation for the training process and measure the machine learning performance using accuracy.

d. Experimental Design

There are several scenarios and parameters used in this experiment. The feature representation uses TF-IDF (term frequency-inverse document frequency), with 5.000 features. The training process is divided into two ways: (1) split of 80% training and 20% testing data; and (2) 10-cross validation approach by looking for the average accuracy for ten training-testing models. The pre-processing is divided into two scenarios: with close-words or removing close-words (rcw) method. We involved the rcw method to prove that close-word can negatively impact the model accuracy. The SVM's kernel uses RBF (Radial Basis Function) and several C's parameters (1, 10, 100). The higher the C value, the smaller the hyperplane, and vice versa. Finally, the experiment is over after the model has done training and testing data.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Machine learning identification performance is measured using accuracy. The accuracy value is obtained based on four components from the confusion matrix table. True Positive (TP) is the value when the machine learning prediction results match the actual data. False Positive (FP) is a value when the machine learning prediction results do not match the factual data. False Negative (FN) is a value when the machine learning prediction is wrong, but the actual data is correct. Finally, True Negative (TN) is the value when the machine learning prediction is wrong, and the actual data is wrong. The accuracy formula can be seen in Equation (1).

3. Result

a. Language Identification Dataset

We use a corpus from Leipzig Corpora Collection (LCC) to help identify the Indonesian or Malaysian language. Table 1 shows the dataset variants based on the size of the number of data instances. The first and second column contains some instances (sentences) in Indonesian and Malaysian language, and the third column is a summation of both column before. We called "ds-ind-msa-10K" dataset if the dataset contains 10.000 Indonesian-Malaysian instances. It means "ds" stands for a dataset, "ind-msa" stands for Indonesian-Malaysian, and "10K" stands for 10.000. We only use the ds-ind-msa-10K dataset, the smallest one (first row in the Table). It is because we have limited resources and time for model testing. The biggest dataset from LLC is two million rows of instance data. We named it ds-ind-msa-2M dataset. We suggested to uses deep learning techniques to process the dataset.

We obtained statistical information from the ds-ind-msa-10K dataset after pre-processing step. The number of tokens (words) for the Indonesian language corpus is 13.208 words, while for the Malaysian language is 12.103 words. Meanwhile, the shortest and longest sentences in the dataset are 4 (letters) and 53 (letters), including spaces.

Figure 2 and Figure 3 shows the number of Indonesian and Malaysian word frequencies in the top-20. Figure 2 focused on the top-20 of similar words of both language, whereas Figure 3 reveal the same and different words in the both language. Based on the data in Figure 2 and Figure 3, we will illustrate that identifying similar languages is very important to be analyze. Thirteen words are the same in both figures, and seven words are different. These are the examples of the similar words: "yang" (which), "dengan" (with), "untuk" (for), and others. The examples of not similar words: "anda" (you), "dapat" (can), "lebih" (more), and others. The word "anda" (you) in Indonesia's top-20 is not listed in Malaysia's top-20.

Table 1. Variants dataset based on the size and the number of instances

LCC-IND	*LCC-MSA*	*DS-IND-MSA*
5.000	5.000	10.000 (10K)
25.000	25.000	50.000 (50K)
50.000	50.000	100.000 (100K)
250.000	250.000	500.000 (500K)
500.000	500.000	1.000.000 (10M)
1.000.000	1.000.000	2.000.000 (10M)

These thirteen words (65%) can be biased during machine learning or deep learning training because the model cannot determine whether the word is Indonesian or Malaysian. Therefore, the similar word in language identification can be used as a challenge and a special issue for machine learning or deep learning techniques. The illustration is only with a tiny bit of data, 20 words. What if the data is more? The more the similar words, the more bias will appear.

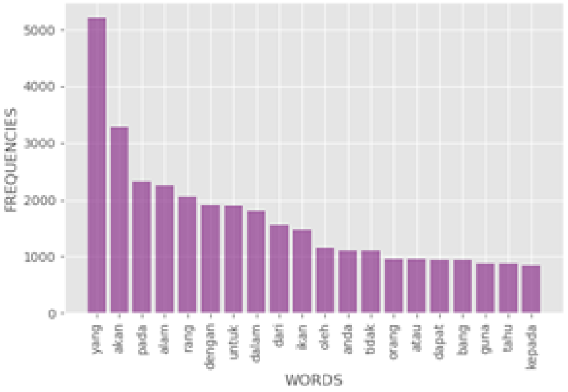
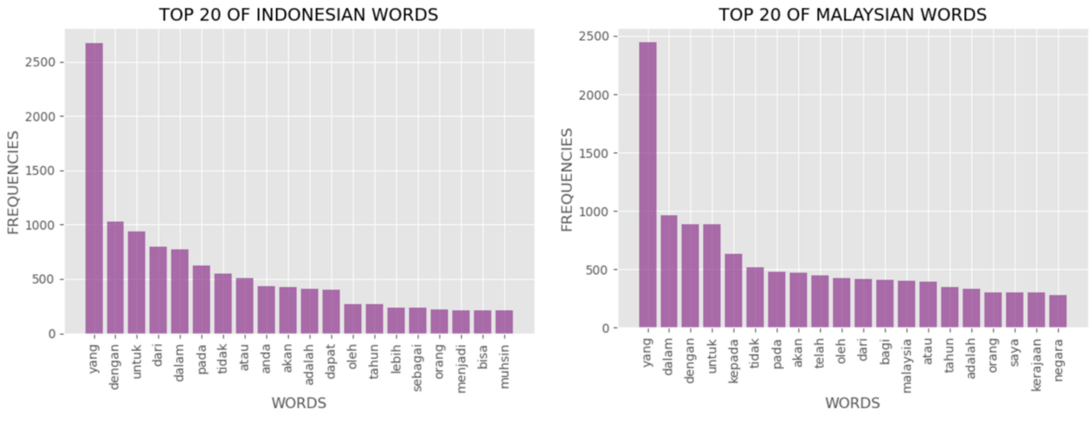


Figure 2. Top-20 of Indonesian and Malaysian Word Frequencies

Also in the Figure 2 and Figure 3, we can draw that high-frequency words are conjunctions, such as the words: "yang" (which), "dengan" (with), "for" (untuk), and others. Some of these words belong in Indonesian and Malaysian. Such a word that belongs in other languages is called close words. Figure 3 shows the word frequency of Indonesian-Malaysian close words in the top-20. We found that there are around 4.683 close words in ds-ind-msa-10K dataset. It means that 46.83% of words in the ds-ind-msa-10K dataset can be biased for machine/deep learning training.



Top-20 of Indonesian-Malaysian close-word

b. Experimental Results

The first experiment uses Naive Bayes Multinomial as baseline model. Table 2 contains The experimental results of Naive Bayes Multinomial (Baseline) and Support Vector Machines (SVM)

There are six experiments divided into three categories: (1) Evaluated the training dataset (80%) with or without rcw (remove close word); (2) Evaluated the testing dataset (20%) with or without rcw; and (3) Evaluated the dataset using k-fold cross-validation (marked by asterisk symbol). The 'train' meant that the result of evaluation using train data. While the 'test' is meant to be evaluation with 20% train data after training the model with 80% train data. In SVM columns, there are three additional columns for accuracy based on C parameter values.

Table 2. Comparison results of NB and SVM

Dataset	rcw	*Naive Bayes* (NB)	*Suport Vector Machines (SVM)*		
		Acc. (%)	Acc. (%), C=1	Acc. (%), C=10	Acc. (%), C=100
Train	-	96.54	99.85	100	100
Train	yes	94.36	94.36	94.36	94.36
Test	-	94.40	93.10	93.05	93.05
Test	yes	93.95	93.35	93.45	93.45
Train*	-	91.35	89.60	89.60	89.60
Train*	yes	89.00	89.00	89.00	89.00

*) using 10-cross validation

Based on the results, we found that the model accuracy using training or testing data is higher when the experiment does not use the remove close-word (rcw) method in pre-processing. All close words will be excluded from the dataset if it uses rcw method. Likewise, for the average accuracy using 10-cross validation. These results indicate that using rcw has less impact or reduces the accuracy value.

The next experiment uses SVM with all parameter values of C (1, 10, 100). The results obtained for C=1 are slightly different from the previous results. Except for the test dataset, all experiments have high accuracy when not using rcw method. In the test dataset, the accuracy of using rcw is higher, with a difference of about 0.25%.

Furthermore, The results obtained for C=10 are similar to the previous experiment. The accuracy value has increased both the training and test data. However, the average accuracy remains the same as the previous test. Finally, The results obtained for C=100 are the same as C=10, and it continued for C=1.000 and so on. The accuracy values will be convergent even though the value of C keeps increasing.

In Table 2 we can compare the accuracy of Naive Bayes Multinomial and the best SVM results (C=10). SVM only excels in the training data. The rest, Naive Bayes exceed SVM. Some scenarios have the same accuracy value for both. For processing speed, the Naive Bayes is superior to SVM. In general, these results indicate that using rcw in pre-processing step has not improved the accuracy for both models.

This study uses simple techniques, but the results unexpectedly reach an accuracy of about 90%. It needs to be analyzed again to determine whether the results tend to be overfitted by using unseen test data that is unrelated to the dataset. This study also processes words numerically using TF-IDF which works at the lexical level. Therefore, the results of this study have not captured the semantic meaning. We can use the n-gram, word2vec, and other techniques.

We also predicted unlabelled data using Naive Bayes Multinomial and SVM. The data used is taken from the Twitter dataset [3] for the top-10 data. Naive Bayes prediction successfully predicts all texts correctly. The assessment is given by human justification. Meanwhile, there is one prediction error in the SVM prediction results. The sentence should be in Indonesian but is predicted as Malaysian. This study has not tested all data from the Twitter dataset, where the total data without labels is 453.390 sentences.

4. Conclusion

The experiment results have proved that the language identification accuracy in specific datasets using Naive Bayes Multinomial and SVM is not too different. Both provide relatively high accuracy, around 90% and above. SVM is superior to Naive Bayes for testing accuracy with training data with a difference of 6.55%. While the accuracy of testing with test data, Naive Bayes is superior to SVM (best) with a difference of 0.95%. The test scenario using remove close word (rcw) in pre-processing has not been able to improve accuracy using either the Naive Bayes or SVM. Testing (prediction) on unlabeled data using the Twitter dataset shows that the Naive Bayes is slightly superior to SVM, but the data tested is only ten from 453,390 data. This study shows that automatic language identification accuracy for Indonesian and Malaysian languages is relatively high even though using simple techniques. However, there is still room for improvement. For example, we can use n-gram or word2vec to replace TF-IDF representation to capture the semantic meaning. In addition, deep learning techniques such as LSTM (Long Short Term Memory) can be used for large data.

Acknowledgement

This research was supported by the Faculty Grant (No. 1718.b/UN9.1.9/LT/2020, 28 July 2020), Faculty of Computer Science, Universitas Sriwijaya (Sriwijaya University), South-Sumatera, Indonesia.

References

[1] T. Jauhiainen, K. Lindén, and H. Jauhiainen, "Evaluation of language identification methods using 285 languages," in *NoDaLiDa 2017 - 21st Nordic Conference of Computational Linguistics, Proceedings of the Conference*, 2017, no. May, pp. 183–191.

[2] S. Carter, W. Weerkamp, and M. Tsagkias, "Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text," *Language Resources and Evaluation*, vol. 47, no. 1, pp. 195–215, 2013, doi: 10.1007/s10579-012-9195-y.

[3] R. Ferdiana, F. Jatmiko, D. D. Purwanti, A. S. T. Ayu, and W. F. Dicka, "Dataset Indonesia untuk Analisis Sentimen," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 8, no. 4, p. 334, 2019, doi: 10.22146/jnteti.v8i4.533.

[4] B. Ranaivo-Malancon, "Automatic Identification of Close Languages - Case study: Malay and Indonesian," *ECTI-CIT*, vol. 2, no. 2, pp. 126–134, 2006, doi: 10.37936/ecti-cit.200622.53288.

[5] Z. Indra, N. Zamin, and J. Jaafar, "A Language Identifier for Indonesian and Malay Text Document," In *2015 International Symposium on Mathematical Sciences and Computing Research (iSMSC)* (pp. 127-131). IEEE. p. 5, 2015.

[6] H. Nomoto, A. Shiro, and S. Asako, "Reclassification of the Leipzig Corpora Collection for Malay and Indonesian." *NUSA* 65 (2018): 47-66.. doi: 10.15026/92899.

[7] Yoav Goldberg, "A Primer on Neural Network Models for Natural Language Processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[8] A. Massaro, V. Maritati, and A. Galiano, "Automated self-learning Chatbot initially built as a FAQS database information retrieval system: Multi-level and Intelligent Universal Virtual Front-Office Implementing Neural Network," *Informatica (Slovenia)*, vol. 42, no. 4, pp. 515–525, 2018, doi: 10.31449/inf.v42i3.2173.

[9] A. Massaro, D. Giannone, V. Birardi, and A. M. Galiano, "An innovative approach for the evaluation of the web page impact combining user experience and neural network score," *Future Internet*, vol. 13, no. 6, p. 145, 2021, doi: 10.3390/fi13060145.

[10] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning Based Text Classification: A Comprehensive Review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–40, 2021.

[11] A. Massaro, V. Vitti, A. Galiano, and A. Morelli, "Business Intelligence Improved by Data Mining Algorithms and Big Data Systems: An Overview of Different Tools Applied in Industrial Research," *Computer Science and Information Technology*, vol. 7, no. 1, pp. 1–21, 2019, doi: 10.13189/csit.2019.070101.

[12] Y. Li and B. Liu, "A new vector representation of short texts for classification," *International Arab Journal of Information Technology*, vol. 17, no. 2, pp. 241–249, 2020, doi: 10.34028/iajit/17/2/12.

[13] E. Tromp and M. Pechenizkiy, "Graph-based N-gram language identification on short texts," in "*Proceedings of the 20th annual Belgian-Dutch Conference on Machine Learning*," 2011, pp. 27–34.

[14] P. Gamallo, M. Garcia, S. Sotelo, and J. R. Pichel, "Comparing ranking-based and Naive Bayes approaches to language detection on tweets," in *CEUR Workshop Proceedings*, 2014, vol. 1228, pp. 12–16.

[15] A. Jaech, G. Mulcaire, S. Hathi, M. Ostendorf, and N. A. Smith, "Hierarchical Character-Word Models for Language Identification," in EMNLP 2016 - *Conference on Empirical Methods in Natural Language Processing, Proceedings of the 4th International Workshop on Natural Language Processing for Social Media, SocialNLP 2016*, 2016, pp. 84–93. doi: 10.18653/v1/w16-6212.

[16] T. Kocmi and O. Bojar, "LanideNN: Multilingual language identification on character window," in *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, 2017, vol. 2, pp. 927–936. doi: 10.18653/v1/e17-1087.

[17] D. Jurgens, Y. Tsvetkov, and D. Jurafsky, "Incorporating dialectal variability for socially equitable language identification," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017, vol. 2, pp. 51–57. doi: 10.18653/v1/P17-2009.

[18] D. Goldhahn, T. Eckart, and U. Quasthoff, "Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages," in *Proceedings of the 8th International Conference on Language Resources and Evaluation*, LREC 2012, 2012, pp. 759–765.

[19] L. Bottou and C.-J. Lin, "Support Vector Machine Solvers," *Large-Scale Kernel Machines*,

vol. 3, no. 1, pp. 301–320, 2007, doi: 10.7551/mitpress/7496.003.0003.

[20] W. S. Noble, "What is a support vector machine?," *Nature Biotechnology,* vol. 24, no. 12, pp. 1565–1567, 2006, doi: 10.1038/nbt1206-1565.

[21] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial naive bayes for text categorization revisited," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2004, vol. 3339, pp. 488–499. doi: 10.1007/978-3-540-30549-1_43.