Automatic Categorization of Tagalog Documents Using Support Vector Machines

April Dae C. Bation
Department of Computer Science
University of the Philippines Cebu
Cebu City, Philippines 6000
acbation@up.edu.ph

Erlyn Q. Manguilimotan
AI Innovation Center
Weather News, Inc.
Chiba, Japan 261-0023
erlynqm@gmail.com

Aileen Joan O. Vicente
Department of Computer Science
University of the Philippines Cebu
Cebu City, Philippines 6000
aovicente@up.edu.ph

Abstract

Automatic document classification is now a growing research topic in Natural Language Processing. Several techniques were incorporated to build a classifier that can categorize documents written in specific languages into their designated categories. This study builds an automatic document classifier using machine learning which is suited for Tagalog documents. The documents used were news articles scraped from Tagalog news portals. These documents were manually annotated into different categories and later on, underwent preprocessing techniques such as stemming and removal of stopwords. Different document representations were also used to explore which representation performed best with the classifiers. The SVM classifier using the stemmed dataset which was represented using TF-IDF values yielded an F-score of 91.99% and an overall accuracy of 92%. It outperformed all other combinations of document representations and classifiers.

1 Introduction

Due to the explosive growth of documents in digital form, automatic text categorization has become an important area of research. It is the task of assigning documents, based solely on its contents, to predefined classes or categories. Through time, approaches to this field of study evolved from knowledge engineering to machine learning. In the machine learning approach, the defining characteristics of each document are learned by the model from a set of annotated documents used as “training” data. Such includes Naïve Bayes and Support Vector Machine classifiers.

Different standard machine learning techniques treat text categorization as a standard classification problem, and thereby reducing the learning process into two steps — feature selection and classification learning over the feature space (Peng et. al., 2003). Of these two steps, feature selection is more critical since identifying the right features will guarantee any reasonable machine learning technique or classifier to perform well (Scott & Matwin, 1999). However, feature selection is language-dependent. Several preprocessing methods such as stopword removal, lemmatization and root-word extraction require domain knowledge of the language used (Peng et. al., 2003).

Methodologies used in researches concerning automatic document categorization are unique from language to language, depending on the structure and morphological rules of the specific language. Although automatic text categorization is becoming a great area of research in most languages aside from English such as Chinese and Arabic, researchers have paid little to no attention

31st Pacific Asia Conference on Language, Information and Computation (PACLIC 31), pages 346–353
Cebu City, Philippines, November 16-18, 2017

in categorizing Tagalog documents. Tagalog exhibits morphological phenomena that makes it a little different than the English language. Thus, this study aims to investigate the factors and explore on different methods that will affect the process of building a Tagalog document classifier. Specifically, this study intends to:

- Collect Tagalog news articles and label them according to their category
- Represent and extract features from documents using NLP techniques
- Build an SVM Classifier
- Evaluate classification performance and present results

2 Related Studies

2.1 Document Categorization and Machine Learning

Different researchers have already explored on automatic document categorization to help manage documents efficiently. Over the years, many approaches have already been adopted to such research problem — from data mining techniques to machine learning models.

Although many approaches have been proposed, text categorization is still a major area of interest since these classifiers have been devoted and focused on English documents and can still be improved.

Several studies used different machine learning models in document categorization. McCallum and Niggam (1998) compared two different types of naïve bayes which assumes that all attributes of the examples are independent of each other. Eyheramendy et. al (2003) used multinomial naïve bayes but found out that it is often outperformed by support vector machines. The use of decision trees for multi-class categorization was explored by Weiss et. al (1999). K-Nearest Neighbors algorithm is also applied in text categorization such as that in a study by Soucy and Mineau (2001) where the model performed better with only few features. Zhang and Zhou (2006) experimented on the use of neural networks for multilabel categorization.

Although there were several researches on document categorization, none had replaced Support Vector Machines as the state-of-the-art method in this research area. A study by Joachims (1998) showed that Support Vector Machines are suited for text categorization, and has consistently showed good performance in all experiments. Yang and Liu (1999) conducted a controlled study and re-examined five of machine learning text categorization methods where SVM outperformed all other methods.

2.2 Support Vector Machines

This type of classifier, proposed by Vladimir Vapnik and Alexey Chervonenkis, began to establish as the state-of-the-art method for text categorization in 1992. Figure 1 shows the framework for SVM on text categorization.

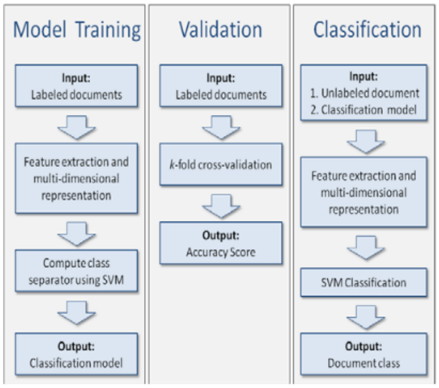


Figure 1. Classification Infrastructure of SVM on Text Categorization (Mertsalov and McCreary, 2009)

Joachims (1998) concluded that SVM will work well for text categorization since (1) it uses overfitting protection which gives it the potential to handle large feature spaces, especially that learning text classifiers deal with more than 10000 features, (2) document vectors are sparse which means that only few entries in it have non-zero values.

2.3 Existing Classifiers in Other Languages

Since feature extraction is language-dependent and requires language-specific knowledge, building a classifier for documents in different languages will introduce different challenges.

In an automatic Arabic document categorizer by Kourdi et. al. (2004), the word morphology was considered. A root extraction technique suited for the non-concatenative nature of Arabic and the challenge of their plural and hollow verbs was used.

In Chinese document categorization, word segmentation became a challenging issue since the language does not have a natural delimiter between words, unlike English and other Indo-European languages. He et. al. (2003) adopted a word-class bigram model to segment each training document into a feature vector.

With regards to Indian languages, Nidhi (2012) stated that using only statistical approaches to classify Punjabi documents won't provide good classification results since the language has a very rich inflectional morphology compared to the English language. This means that there is a need of linguistic approaches and a good understanding of the language's morphology for the selection of the features that will increase efficiency. Nidhi (2012) used a rule-based approach to extract language-dependent features.

Concerning Tagalog, no work has been done to classify Tagalog documents. Although recently, there are morphological analysis tools for the Tagalog language such as the Two-level Engine for Tagalog Morphology (Nelson, 2004), the Tagalog Stemming Algorithm (TagSA) (Bonus, 2012), different proposed POS taggers including the works of Cheng (n.d.) and Reyes et al., (2014), none of which are being applied in the automatic categorization of Tagalog documents.

3 Methodology

This study follows the basic framework for document categorization which is divided into three, namely: data preparation and preprocessing, feature extraction and selection, and the building of classifier.

3.1 Preprocessing of Data

In the preprocessing of data, the first step was removing the whitespaces and punctuations. The documents were also transformed into lowercase.

In the next step, stopwords were removed. This includes words such as *ang, mga, si, dahil,* etc. These are frequent occurring words in Tagalog language which do not offer information about the category of the document.

Lastly, stemming was done. This is used to reduce the words in the documents into its canonical form. Words with the same canonical form is counted as one. For example, *maaga*, *pinakamaaga*, and *umaga*, will be counted as one since they all have the same canonical form, *aga*.

In Tagalog, there are four types of affixation: (1) prefixation, (2) infixation, (3) suffixation, and (4) circumfixation. Prefixation is when the bound morpheme is attached before the root word, infixation is when it is attached within the root word, and suffixation is when it is attached at the end. Circumfixation is when the bound morpheme can occur as prefix, infix, or suffix. Reduplication of these affixes is also common in the language. The stemmer created by the researcher was meant to remove the affixes, including the reduplicated parts, and retrieve the root word only

The stemmer retrieves the canonical form by removing all affixes that can occur as prefix, infix, and suffix. Affixes in Tagalog include *um*, *ma*, and *in*. Words with these affixes include *k(um)ain, (ma)bilis, s(in)abi.* After stemming these words, *kain, bilis* and *sabi* will be retrieved respectively.

The stemmer also removes reduplicated parts. In the word *pupunta*, the morpheme *pu-* was reduplicated; hence it will be removed. After stemming, its canonical form, *punta,* will be retrieved.

On the other hand, Non-Tagalog words were considered foreign words. -

3.2 Document Representation and Feature Extraction

After the preprocessing method, a Bag-of-Words model, containing all words in the documents, was created. This is used as the basis for extracting features.

3.3 Feature Vectorization

Typically, the feature space consists of an $m \times n$ matrix where m is equal to the number of documents and n is equal to the number of tokens in the Bag-of-Words.

In this study, three schemes in numerical representation were used, namely: Binary Representation, Word Counts, and the TF-IDF.
–

3.4 Classification of Documents

After vectorizing the documents into different numerical representations, they were then shuffled and divided into two: the training set and testing set. 80% of the dataset went to the training set while the remaining 20% went to the testing set. Sklearn's train_test_split was used.

In this study, two classifiers were experimented, namely: Naïve Bayes and Support Vector Machines. Both were implemented using Python's sklearn.

Support Vector Machines
In this study, a linear kernel and a one-vs-all strategy were used where a single classifier per class is trained, with the samples of that class as positive samples and all other samples as negatives The OneVsRestClassifier, together with the LinearSVC of sklearn were utilized.

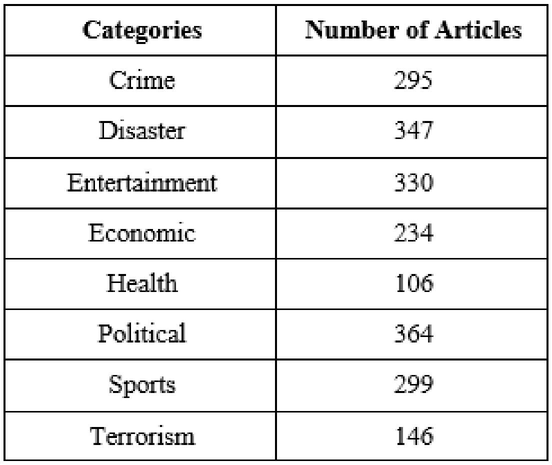
Multinomial Naïve Bayes
For the second classifier in this study, a Multinomial Naive-Bayes, which estimates probabilities of a given document to belong to a specific category, was used. The MultinomialNB of sklearn was used in this study.

4. Results and Discussions

Several experiment setups with the different document representations and machine learning classifiers were conducted. Out of the 2,121 news articles, 1,696 news articles (80%) went to the training set. The remaining 425 news articles (20%) went to the testing set.

4.1 Dataset

The dataset is comprised of Tagalog news articles retrieved from Philippine news websites from August 2016 to January 2017 using scrapy (https://scrapy.org/). The collected data comprised of 2,121 manually annotated news articles. Table 1 summarizes the distribution of data for each category.



Categories	Number of Articles
Crime	295
Disaster	347
Entertainment	330
Economic	234
Health	106
Political	364
Sports	299
Terrorism	146

Table 1. Distribution of Pre-defined Categories

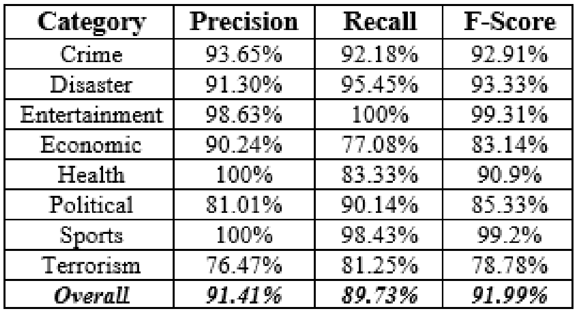
4.2 Document Representation

In this study, three document representations were used for the experiments that were conducted — Binary Feature Representation, Word Count Representation, TF-IDF Representation. From the training set, 22,824 total terms/words were retrieved and stored in the Bag-of-Words.

Some words included in the Bag-of-Words are not part of the Tagalog vocabulary. These includes frequently occurring foreign words and proper nouns such as are *city* and *duterte*. Some proper nouns were also stemmed such as *philippe* which is originally *philippine* but -in- was removed because the stemmer thought it is an infix.

4.3 Core Experiment

For the core experiment, an SVM classifier is used together with the TF-IDF representation for all documents. The overall accuracy of this classifier is 92%. Table 2 summarizes the performance metrics of the classifier.



Category	Precision	Recall	F-Score
Crime	93.65%	92.18%	92.91%
Disaster	91.30%	95.45%	93.33%
Entertainment	98.63%	100%	99.31%
Economic	90.24%	77.08%	83.14%
Health	100%	83.33%	90.9%
Political	81.01%	90.14%	85.33%
Sports	100%	98.43%	99.2%
Terrorism	76.47%	81.25%	78.78%
Overall	***91.41%***	***89.73%***	***91.99%***

Table 2. SVM Classifier Performance

Based on Table 2, the classifier was able to yield relatively high F-Scores, except that of Terrorism which yielded an F-Score of only 78.78%. This was expected since the amount of news articles that belong to this category was relatively low compared to that of other categories. On another note, it can be seen in the table that the Entertainment category got a recall of 100%, Health and Sports categories both got a precision of 100%. Also, Economic and Political categories both got an F-score below 90%. This could stem from the nature of the two categories — both talk about the government or the status of the country, which makes it hard for the classifier to distinguish the difference between the two.

4.4 Validation and Evaluation

Based on the core experiment, the performance measure of the classifier is already acceptable. To ascertain the contribution of Tagalog language processing in the classification of Tagalog document, the following experiments were conducted:-

Effect of Stemmer

To show the contribution of stemming to the whole process of building the classifier, an unstemmed dataset was fed to the SVM classifier.

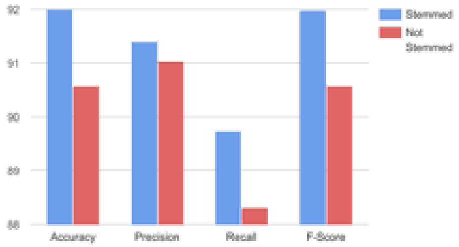


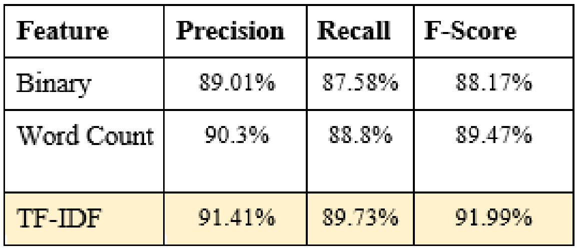
Figure 2. Comparison of Performance Measures for Stemmed and Unstemmed Data

As seen in Figure 2, the classifier with the stemmed data performed better than that with unstemmed data. Although the stemmer wasn't perfect, the process of reducing words to their word stems has helped significantly in improving the performance of classifier.

A Multinomial Naïve Bayes (MultiNB) classifier was tested to see if stemming data still achieves high performance, like in the SVM classifier. Both datasets were fed to the MultiNB classifier. Using TF-IDF, the classifier with the stemmed data yielded an F-Score of 83.55% while the other yielded only 81.41%.

Effect of Document Representation

Based on the previous experiments, it can be seen that TF-IDF representation yielded impressive performance measures for the SVM classifier. For comparison purposes, two other document representation were used — Binary Representation and Word Count.

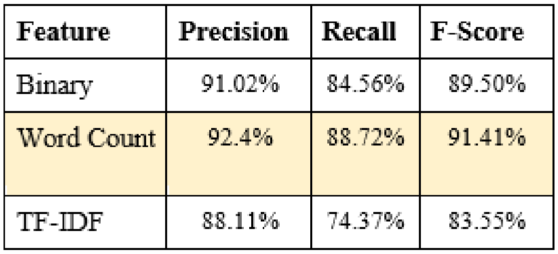


Feature	Precision	Recall	F-Score
Binary	89.01%	87.58%	88.17%
Word Count	90.3%	88.8%	89.47%
TF-IDF	91.41%	89.73%	91.99%

Table 3. SVM Classifier Performance Measure for Different Document Representations

Table 3 summarizes the performance measures of the SVM classifier for the three different document representations where TF-IDF resulted to the highest F-Score of 91.99%.

For the sake of comparison, all three document representation were fed to the MultiNB classifier.



Feature	Precision	Recall	F-Score
Binary	91.02%	84.56%	89.50%
Word Count	92.4%	88.72%	91.41%
TF-IDF	88.11%	74.37%	83.55%

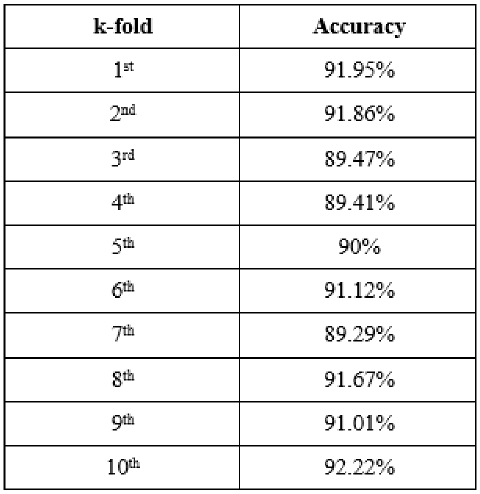
Table 4. Naïve Bayes Classifier Performance Measure for Different Document Representations

Table 4 shows the performance measures for the MultiNB classifier. It can be seen that, unlike in SVM, TF-IDF yielded the lowest F-Score of 83.55% while Word Count yielded 91.41%. The Multinomial Naive Bayes implements the Naive Bayes algorithm for multinomially distributed data, which means that it models the data based on probability counts. Since multinomial distribution normally requires integer feature counts, TF-IDF representation is likely to produce poor results.

Furthermore, TF-IDF with SVM yields a higher F-score compared to that of Word Count with Naïve Bayes, and it in fact outperformed all other combinations of document representation with the classifiers.

Cross-Validation

A 10-fold cross validation scheme was used to validate the performance of the multinomial SVM classifier. Training and testing were repeated 10 times on stratified folds for the whole dataset. Table 5 summarizes the result of the performance of all categories averaged at each fold.



k-fold	Accuracy
1st	91.95%
2nd	91.86%
3rd	89.47%
4th	89.41%
5th	90%
6th	91.12%
7th	89.29%
8th	91.67%
9th	91.01%
10th	92.22%

Table 5. Ten-Fold Cross Validation

The ten-fold cross validated classifier yielded an average accuracy of 90.8%. The test shows that although randomness was introduced to the experiment by means of the folds, the performance is generally the same.

5 Conclusion and Recommendations

5.1 Conclusion

Tagalog document categorization, like in other languages, is affected by many factors. Such includes the size of the corpus, the classifier type, the feature selection and feature reduction method, and the weighting scheme. In this study, stemming each document, representing it with TF-IDF values and using it to train an SVM classifier yielded the highest F-Score of 91.99% among all other combination of methods and experiment setups.

Although the stemming process wasn't perfect, it still served the purpose of conflating and integrating different word forms into their common canonical form; therefore, reducing the number of terms in the whole corpus. This method in computational linguistic can result to either poor or good performance, depending on some cases. In this study, it was shown that stemming, which performs iterative affix removal, is effective in Tagalog documents and that it has contributed to the high performance of the machine learning classifier which automatically classifies documents into categories.

In this study, it was also proven that an SVM classifier performs well in categorizing text data. More than 10000 features were used in this study and each document vector was sparse; however, the SVM classifier was able to handle the large feature space.

5.2 Recommendations for Future Work

Although high performance measures were achieved in building a machine learning classifier that can automatically categorize text documents, it would be better to use a larger dataset with a more even distribution for each class. Future researches could also experiment on more complicated feature representations such as the use of POS tags or N-grams to explore more on their performance on Tagalog documents. Also, researches could try on the use of lemmatization instead of just stemming the Tagalog words. In this research, Tagalog words that weren't stemmed properly by the stemmer, such as *nam* and *sabg*, were included. While stemming only chops off morphemes in words to remove the derivational affixes, lemmatization refers to the use of a vocabulary and morphological analysis of words to be able to do return the correct base or dictionary form of a word. More categories can also be incorporated; for example, Sports can be divided into more specific categories such as Basketball, Volleyball, etc. Lastly, Future researches should also be able to build a classifier that can label the Tagalog documents with more than one category (multi-labeled instead of just multiclass).

References

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management* - CIKM '98. doi:10.1145/288627.288651

Eyheramendy, S., Lewis, D. D., & Madigan, D. (2003). On the naive bayes model for text categorization.

He, J., Tan, A., & Tan, C. (2003). On Machine Learning Methods for Chinese Document Categorization. Applied Intelligence, 18, 311-322. Retrieved from https://www.comp.nus.edu.sg/~tancl/publications/j2003/he03apin.pdf

Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer Berlin Heidelberg.

Kourdi, M. E., Bensaid, A., & Rachidi, T. (2004). Automatic Arabic document categorization based on the Naïve Bayes algorithm. *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages* -Semitic '04. doi:10.3115/1621804.1621819

Lewis, D. (1992). Feature selection and feature extraction for text categorization. In *Proceedings of a Workshop on Speech and Natural Language Processing*, (pp. 212-217). San Mateo, CA: Morgan Kaufmann.

McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, pp. 41-48).

Nelson, Hans J., "A Two-level Engine for Tagalog Morphology and a Structured XML Output for PC-Kimmo" (2004). All Theses and Dissertations. Paper 133. Retrieved from http://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=1132&context=etd

Nidhi, V. G. (2012). Domain Based Classification of Punjabi Text Documents using Ontology and Hybrid Based Approach. *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing* (SANLP) (pp. 109-122). Retrieved from http://www.aclweb.org/anthology/W12-5009

Peng, F., Schuurmans, D., & Wang, S. (2003). Language and task independent text categorization with simple language models. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* - NAACL '03. doi:10.3115/1073445.1073470

Roxas, R. (1997). Machine Translation from English to Filipino: A Prototype. *International Symposium of Multilingual Information Technology (MLIT '97),* Singapore.

Scott, S., & Matwin, S. (1999). Feature Engineering for Text Classification. In Proceedings of the Sixteenth International Conference on Machine Learning (ICML '99), pp. 379- 388.

Soucy, P., & Mineau, G. W. (2001). A simple KNN algorithm for text categorization. In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on (pp. 647- 648). IEEE.

Weiss, S. M., Apte, S., Damerau, F., Johnson, D. E., Oles, F. J., Goetz, T., and Hampp, T. (1999). Maximizing Text-Mining Performance. IEEE Intelligent Systems, 14, 63-69.

Yang, Y., & Liu, X. (1999). A Re-examination of Text Categorization Methods. Carnegie Mellon University. Retrieved from http://www2.hawaii.edu/~chin/702/sigir99.pdf

Zhang, M., & Zhou, Z. (2006). Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. IEEE Transactions on Knowledge and Data Engineering, 18(10), 1338-1351. doi:10.1109/tkde.2006.162