Composer Classification of Filipino Song Lyrics Using Machine Learning

Oliver Isaac L. Chan, Patrick Joseph M. Sadornas, Rafael A. Cabredo, and Charibeth K. Cheng
De La Salle University
Taft Avenue, Manila
oliver_chan@dlsu.edu.ph, patrick_joseph_sadornas@dlsu.edu.ph,
rafael.cabredo@dlsu.edu.ph, chari.cheng@delasalle.ph

ABSTRACT
We implemented a machine learning technique using artificial neural network to perform composer classification (authorship attribution) of songs with lyrics primarily written in Filipino. We used features based on function words, character n-grams, and song-specific features extracted from 98 song lyrics written by three composers and used a multilayer perceptron to model the learning algorithm for automatically classifying song lyrics according to its likely composer. Compared to classification of longer literary materials such as novels, author attribution of short texts such as Filipino song lyrics is generally a more difficult machine learning task. By combining the function words and character n-grams, we can achieve an average classification accuracy of 81.02%. In contrast, a random guess on a song's composer would have an average of 33.33% chance of correct classification. Our results demonstrate a successful application of author attribution methods to short Filipino texts.

Categories and Subject Descriptors
I.5 [**Pattern Recognition**]: Implementation of machine learning methods for classification tasks.

General Terms
Algorithms, Design, Experimentation, Languages.

Keywords
author attribution, neural networks, filipino song lyrics, function words, character n-grams

1. INTRODUCTION
Statistical or computational methods for authorship attribution rely on measurement of some textual features that facilitate the differentiation of texts written by different authors. While authorship attribution studies were conducted as early as the 19th century, the detailed study of Mosteller and Wallace in 1964 on the authorship of the "Federalist Papers" is considered the most influential work in authorship attribution. Their work initiated the non-traditional authorship attribution as compared to traditional human-expert based methods. [13]

Aside from traditional application to literary research, authorship attribution has various potential applications as cited in [13] which includes intelligence-gathering (attribution of messages or proclamations to known terrorists) [1], criminal law (identifying writers of harassing messages and verifying the authenticity of suicide notes) [4], and civil law (copyright disputes) [7], and computer forensics (identifying authors of source code of malicious software) [6]. Research on authorship attribution in the recent decade has focused less on disputed literary works and more on efforts to develop effective methods and practical applications dealing with real-world texts such as those found on the Internet [13].

As cited in [12], previous research on author identification point out that classification becomes more difficult as the number of words in the text is reduced [11]. Song lyrics generally fall under the category of short texts and therefore pose a more difficult problem compared to longer literary materials such as novels or essays.

In this paper, we describe the extraction of lexical, character sequence, as well as song-specific features from 98 Filipino songs written by three composers. We present the results of our experiments in individually using and combining the feature sets in a machine learning algorithm for classifying the author of the song lyrics.

While the machine learning task described in this paper focuses on a literary work, the techniques used may also be extended to other tasks involving short texts written in Filipino. We used attribution techniques that are typically applied to poems and other short texts.

2. RELATED WORK
One way of representing a piece of text is as a sequence of tokens (e.g. words) grouped into sentences and using a vector of word frequency is the simplest approach [13]. In particular, function words (words that are not nouns, verbs or adjective) are common lexical feature used for author attribution. Function words have been shown to be a superior feature for distinguishing between authors using machine learning techniques [2]. Authors tend to use function words without even noticing that they are using them in a certain pattern [13].

Character n-gram is another computationally simple feature that is able to capture the nuances of style, including lexical information [5]. As described in [9], using all possible n-grams as features is generally impractical as it leads to an exponential increase in features on higher orders of n. Thus, they proposed using a cut-off based on the relative frequency of an n-gram in a given data set to reduce the number of n-grams to be used as features for machine learning.

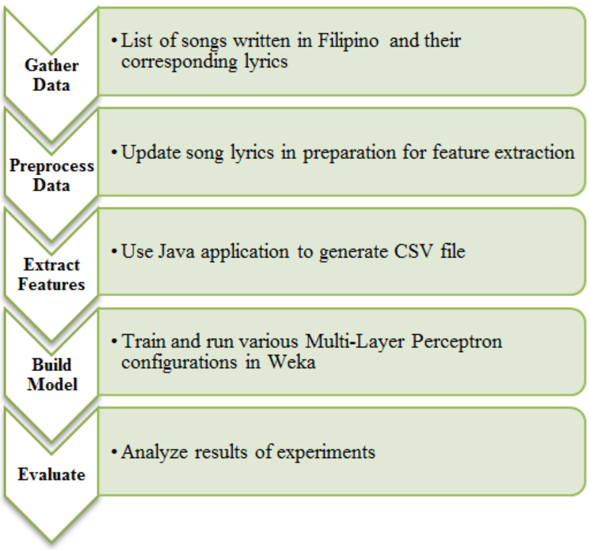
The use of neural networks has been shown to be useful for authorship identification and for classification problems in general [9, 5]. As briefly described in [5], a neural network is built from nodes connected by weighted links. It has an input layer corresponding to the input features used and an output layer that correspond to the desired classifications. It may contain several hidden layers. A node's activation function uses the weighted sum of the input nodes to determine its output. The weights between nodes are adjusted during training until the classification error rate is reduced, usually through a method called gradient descent. The weights of the hidden units are usually computed using a method known as backpropagation. Using this supervised learning method, a training instance is presented to the network and its output is compared to the actual class label. The weights are subsequently adjusted to bring the output of the network closer to the actual class label. After training the network, instances not included in the training data may be presented to the network which will predict the corresponding class label.

Based on our search for related literature, we were not able to find a published work specifically performing authorship attribution of Filipino song lyrics or poems. However, we found an authorship attribution study of opinion articles (which are also relatively short texts) written by Filipino columnists [12]. The researchers used a modified word frequency ranking as their primary feature for linear discriminant analysis.

3. DATA SET AND METHODOLOGY

Our authorship attribution task focused on the songs written by Alfonso Miranda, Jr. ("Chito"), Eleandre Basiño Buendia ("Ely") and Rico Blanco. The fact that the composers chosen were of the same gender and generally belong to the same music genre adds difficulty to the classification task. The following figure summarizes the major steps we undertook in this study.

Figure 1. Key Steps Performed



We assigned composer labels to the lyrics based on publicly available information and validated this using information obtained from Filipino Society of Composers, Authors, and Publishers (FILSCAP). We requested the list of songs composed by the three artists from FILSCAP and a raw list that included both English and Filipino songs was provided. Songs primarily written in Filipino were selected from the said list and the corresponding lyrics of these songs were obtained from the Internet. As these artists sometimes include English words or lines in their work, song lyrics were included in the data set as long as it had more lines written in Filipino than in English. We identified a total of 98 Filipino songs written by the three artists.

Inaccuracies noted in the lyrics obtained were manually corrected and instructive texts (e.g. "repeat chorus") were replaced with the relevant lyrics of the song.

The updated lyrics were used as input to the feature extraction application that we developed in Java. The application generated a file in CSV format containing the artist (as class label), song title, and various features.

Weka data mining software [8] was used to build the classifier. We used artificial neural network, specifically a multilayer perceptron (MLP), to build a machine learning model using the features extracted from the song lyrics. We experimented with various parameter settings for the MLP in Weka and analyzed the results to determine the best performing set of features and MLP configuration.

The number of songs was fairly distributed between the three artists. However, having only 32 or 33 song for each artist imposed a limit on the amount of training data. Thus, we employed 10-fold cross-validation for our experiments instead of further dividing the data set into a fixed number of training set and test set.

4. FEATURES USED

4.1 Lexical Features

Function words are common lexical features used for author attribution because they scale well across different topics, context and possibly genre. Because function words are topic-independent, it is a decent general feature to cover different compositions across different composers.

For this experiment, we used 277 distinct Filipino function words obtained from the Automated English and Filipino Lexicon Builder (AEFlex) System [10] of the De La Salle University. We have created and tested different sets of features from our set of function words.

4.1.1 Function word frequency

A common representation for function words is a simple count vector where the number of occurrence for each of the 277 function words is tallied. The result would give a 277-element vector of numerical data for each composition.

4.1.2 Function word average frequency (normalized by number of tokens)

In order to scale and normalize the count vector, we took the average occurrence of each function word then divided the total count with the number of tokens in a composition.

4.1.3 Function word average frequency (normalized by number of function words)

Another version for the averaged function words, we computed for the average of the function words over the total number of function words in a composition.

4.1.4 Function word relative frequency (normalized by count of the most frequent function word)

This feature attempts to capture two attributes in a composition: 1) existence of a certain function word and 2) rank of a function word relative to the most frequently-occurring function word in that composition. The number of occurrence for each function word is divided by the count of the highest occurring function word in the composition.

4.1.5 *Function word binary vector*
For this feature, the frequency of occurrence of each function word was disregarded and instead used a binary value to flag if a certain function word exists in a composition. If a certain function word exists, its corresponding element in the vector is set to 't' (true), otherwise is set to 'f' (false). The resulting feature is a 277-element binary vector with 't' or 'f' values.

4.1.6 *Function word ranked nominal*
This feature is an extension of the function word binary vector described above. Instead of just checking for existence, this feature ranks the function words according to their number of occurrence. To illustrate, if "ang" is the function word with the highest count on a particular composition, the attribute "Rank 1" is set with the value "ang". After all the 277 function words are ranked, this feature set would result in a 277-element nominal vector. If a function word has a count of zero (which means it does not exist in composition), the word "EMPTY" is set as the value for that attribute instead of the function word itself. Using this feature, we were able to capture the number of distinct function words used in a particular composition.

4.2 Character N-Grams
In our implementation of character n-grams, we replaced spaces between words with an underscore symbol and converted all text in lowercase. For example, the phrase "awitin ko" would yield the following character 3-grams: "awi", "wit", "iti", "tin", "in_", "n_k", and "_ko".

Some of the resulting features will also unlikely to be used as they are statistically improbable to occur in a text (e.g. "qqq", "qqr", "qqs", and so on). Thus, we extracted the most frequently used character n-grams from the entire data set and used as features those that have a relative frequency (normalized by the total number of n-grams in the entire data set) greater than 0.04% as done in [9] and 0.02%. The lower threshold was set to determine possible improvement in performance as a result of increasing the number of features. This resulted to the following number of features for character n-grams of order 2, 3, and 4:

Table 1. Number of Character N-Gram Features

	n=2	n=3	n=4
Relative frequency >0.4%	67	37	12
Relative frequency >0.2%	97	101	49

We performed our experiments using the neural network classifier of the Weka tool, which is called a multilayer perceptron.

4.3 Song-Specific Features
We also considered several song-specific features to determine their potential use in discriminating between the three artists. These features relate to the song structure as well as to the occurrence of certain words within the song.

4.3.1 *Stanza count*
This feature is the number of stanza (i.e. verses and chorus) that a particular song has.

4.3.2 *Average syllables per line*
This feature represents the average number of syllables per line in a song.

4.3.3 *Title count*
For this feature, the number of times that the song title occurs in the song lyrics is counted.

4.3.4 *Distinct words used*
This is the count of unique words used within the song and can be viewed as a simple measure of vocabulary richness.

5. EVALUATION
We used the percentage of correctly classified instances as well as the kappa statistic (κ) to evaluate the reliability of the machine learning methods and features used in the experiments. Kappa is widely used in the field of content analysis and has been a recommended statistic in evaluating classification tasks in computational linguistics [3]. The kappa coefficient measures pair-wise agreement among a set of coders making category judgments, correcting for expected change of agreement. It is computed using the formula:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \qquad (1)$$

P(A) is the proportion of times that the coders agree and P(E) is the proportion of times that we would expect them to agree by chance. When there is no agreement other than that which would be expected by chance, K is zero. When there is total agreement, K is one. It is possible, and sometimes useful, to test whether or not K is significantly different from chance, but more importantly, interpretation of the scale of agreement is possible [3].

For our task of discriminating between three possible song composers, the P(E) is 1/3. A kappa statistic greater than zero would suggest the machine learning algorithm and the features used have predictive value for the classification task.

6. RESULT OF EXPERIMENTS
6.1 Function Words
The different feature sets based on function words were tested using the multilayer perceptron. We tried several configurations for the perceptron and picked the configuration with the best performance for each set. Figures 2 and 3 show the result for each of the feature set tested. The set "Function Word Ranked Nominal" produced the best result among the function word features. This feature was developed from an analysis of the most common words used by the composers (Table 2).

Figure 2. Kappa Statistic - Function Words

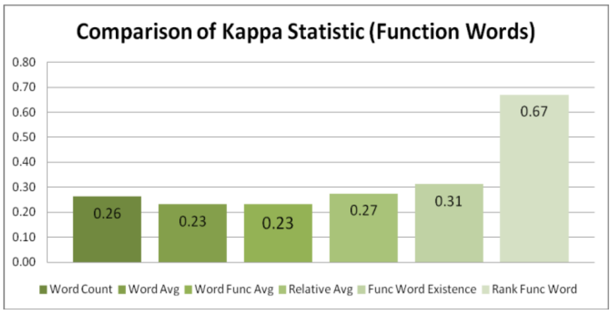


Figure 3. Percentage Correctly Classified - Function Words

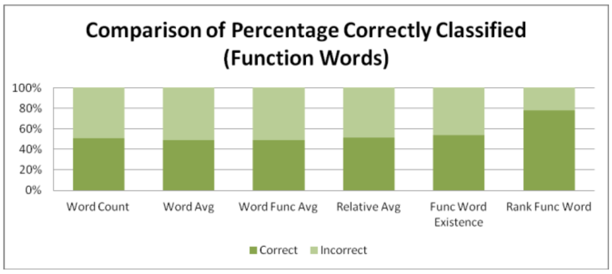


Table 2 below was generated to examine poor results obtained from preliminary use of word n-grams as features. Most of the words that appear on the list are function words. It was also noted that content words appearing on the list have high frequency count primarily due to having one or more songs repeatedly using the same word as part of the song's theme. As the content words rarely recur in different songs, it may not be ideal to use them as features. The ranking of the function words, however, was used as it appeared to differentiate between artists.

Table 2. Top 20 Words Used per Composer

Rank	Chito M	Ely B	Rico B
1	na - 209	na - 157	na - 156
2	hoy - 118	ang - 144	sa - 136
3	ang - 109	sa - 117	ng - 130
4	sa - 91	mo - 81	ang - 101
5	ng - 79	ko - 77	mo - 61
6	ko - 75	ay - 71	at - 50
7	mo - 65	at - 69	ka - 50
8	lang - 62	ng - 69	kung - 41
9	ako - 61	ka - 66	isang - 35
10	ay - 59	lang - 51	hindi - 33
11	ka - 51	lab - 48	kita - 29
12	at - 48	ako - 41	tayo - 29
13	hindi - 48	di - 40	lang - 28
14	pa - 48	la - 33	lahat - 25
15	kong - 38	walang - 30	ayus - 24
16	naman - 36	kung - 29	mong - 23
17	ba - 32	pa - 28	wag - 23
18	di - 32	hindi - 26	langit - 21
19	kung - 29	may - 26	walang - 21
20	kang - 26	ba - 25	may - 20

A model was built using a multilayer perceptron with one 3-unit hidden layer. The training time for the model is set to 50 iterations. The features were filtered using Weka's "Remove Useless" to reduce the number of useless attributes. From 277, the number of features was reduced to 50 for this set. Nominal values were retained as is and were not converted to binary word vector.

The model was tested for 5 times using 10-fold cross-validation. Table 3 shows an average kappa statistic of 0.67 was achieved through this experiment. The lowest kappa statistic was at 0.59 and the highest at 0.72. The average percentage of correctly identified instances was at 77.96% and average percentage of incorrectly identified instances was at 22.04%.

The Function Word Rank Nominal feature is designed to capture the following features: existence of a certain function word, number of distinct function word and rank-based comparison of how frequent a certain function word is used in the lyrics of a particular song.

Table 3. Function Word Ranked Nominal Test Results

Test no.	Correct	Incorrect	Kappa Statistic
1	72.45%	27.55%	0.59
2	81.63%	18.37%	0.72
3	78.57%	21.43%	0.68
4	77.55%	22.45%	0.66
5	79.59%	20.41%	0.69
Average	**77.96%**	**22.04%**	**0.67**

The result of the Function Word Rank Nominal feature set is significantly higher compared to the results from using the other feature sets. However, it is important to note that this feature set requires some manual preprocessing. For each rank feature, function words that were not encountered during feature creation were removed from the nominal set. Since the features of this set are entirely composed of non-normalized nominal values, this feature is prone to bias depending on the order of the nominal values.

Based on analysis of the results of 5 different 10-fold cross-validations and by checking the classification of the instances one by one, we noticed the songs listed in Table 4 were consistently classified incorrectly by the Function Word Rank Nominal feature set.

Table 4. Function Word Misclassified Song Lyrics

Composer	Song Lyrics
Chito M.	Sorry Na
Chito M.	The Crush
Chito M.	Wag Mo Na Sana
Ely B.	Easy Ka Lang
Ely B.	Ang Huling El Bimbo
Ely B.	Harana
Ely B.	Kailan
Ely B.	Kaliwete
Ely B.	Maskara
Ely B.	Minsan
Ely B.	Overdrive
Ely B.	Pare Ko
Ely B.	Sa Wakas
Ely B.	Spoliarium
Ely B.	Walang Nagbago
Ely B.	Wanted Bedspacer
Rico B.	Alab ng Puso
Rico B.	Basketbol
Rico B.	Isang Bandila
Rico B.	Kagat ng Lamok
Rico B.	Panahon Na Naman

For our experiments, song lyrics are considered consistently misclassified if it is incorrectly classified in at least 3 out of the 5 10-fold cross-validation tests. Based on such criterion mentioned, the lyrics listed in the table above are judged to be lyrics that cannot be classified correctly by the Function Word Rank Nominal set. It was noticeable that most errors occurred for lyrics composed by composer Ely Buendia and the least amount of error occurred in lyrics composed by composer Chito Miranda.

6.2 Character N-Grams

The various character n-gram features were used as input of a multilayer perceptron. According to [13], an important issue of the character n-gram approach is the definition of n (i.e. how long the strings should be). A large n would better capture lexical and contextual information, but it would also capture thematic information. Likewise, a large n would substantially increase the number of features. On the other hand, a small n (i.e., 2 or 3) would be able to represent syllable-like information, but it may not capture contextual information.

Based on our preliminary experiments using a multilayer perceptron set to training time of 50 iterations across various number of hidden units (5-45 HU), it was determined that the trigrams and bigrams performed significantly better than the quadgrams as summarized in the table below.

Table 5. Comparison of Baseline N-Gram Performance

	n=2	*n=3*	*n=4*
Highest Kappa Statistic	0.36	0.35	0.15
Highest % Correctly Classified	57.56%	56.82%	43.51%

Based on the preliminary results, the bigrams and trigrams appeared to be better features than the quadgrams. To determine possible performance improvement using the bigram and trigram feature set, the number of hidden units of the neural network as well as the training time was varied to determine the configuration that will perform best. The result of the performance comparison for 37 trigrams (Figure 4) and 97 bigrams (Figure 5) are shown below.

Based on the figures shown, the best performing configuration was for the trigrams using a multilayer perceptron with 20 hidden units set and a training time et to 800 iterations. This resulted to a kappa statistic of 0.39 and correctly classified instance of 59.4%.

Figure 4. Various Neural Network Configurations - Trigrams

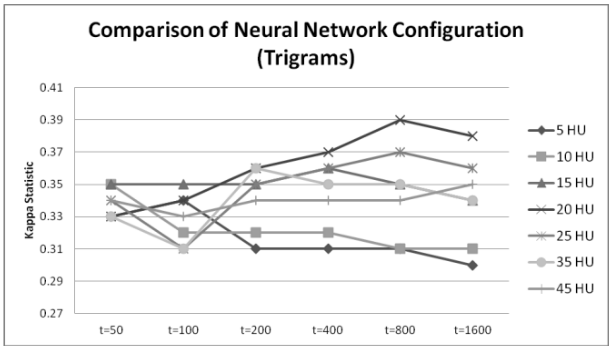
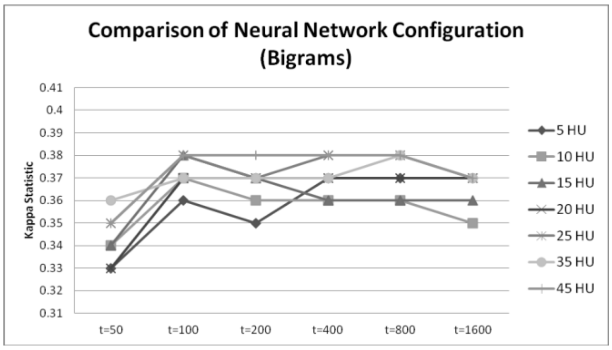


Figure 5. Various Neural Network Configurations - Bigrams



6.3 Song-Specific Features

The four song-specific features were used together as input of a multilayer perceptron with one 3-unit hidden layer. The default training time of 500 iterations was used. The model was likewise tested 5 times using 10-fold cross-validation and resulted in an average Kappa Statistic of 0.22. The average percentages of correctly identified and incorrectly identified instances were at 47.76% and 52.24%, respectively.

Table 6. Song-Specific Features Test Results

Test no.	Correct	Incorrect	Kappa Statistic
1	50.00%	50.00%	0.25
2	47.96%	52.04%	0.22
3	44.90%	55.10%	0.17
4	44.90%	55.10%	0.17
5	51.02%	48.98%	0.27
Average	**47.76%**	**52.24%**	**0.22**

The poor performance of the feature set may be attributed to the lack of significant variability in the features between artists as well as to the fact that the artists selected belong to the same musical genre which may have certain conventions in song structure.

6.4 Combination of Feature Sets

Based on the individual analysis of the results of the features sets, we explored the idea of combining the first two in order to arrive at a better classification performance. The song-specific features were no longer considered due to the low performance results obtained in the initial round of experiments. Table 7 below is an updated version of Table 4. It compares the song lyrics that were misclassified by the Function Word Rank Nominal Set against the classification results of both the Character Bigram Set and the Character Trigram Set.

Based on the following table, we hypothesized that by combining the Function Word Rank Nominal Set, Character Bigram and Character Trigram into a single multi-layer perceptron might correctly classify the songs that were marked as “Yes” for both Bigram and Trigram. Moreover, there might be a possibility to correctly identify lyrics that are marked “Yes” for either bigram or trigram. And by doing so, the combined features might boost the precision of our best performing model.

Table 7. Function Words Misclassified Song Lyrics vs. N-gram Results

Composer	Song Lyrics	Ok with Bigram?	Ok with Trigram?
Chito M.	Sorry Na	**Yes**	**Yes**
Chito M.	The Crush	No	Yes
Chito M.	Wag Mo Na Sana	Yes	No
Ely B.	Easy Ka Lang	Yes	No
Ely B.	Ang Huling El Bimbo	Yes	No
Ely B.	Harana	**Yes**	**Yes**
Ely B.	Kailan	No	No
Ely B.	Kaliwete	No	Yes
Ely B.	Maskara	No	Yes
Ely B.	Minsan	No	Yes
Ely B.	Overdrive	No	No
Ely B.	Pare Ko	No	No
Ely B.	Sa Wakas	**Yes**	**Yes**
Ely B.	Spoliarium	No	No
Ely B.	Walang Nagbago	**Yes**	**Yes**
Ely B.	Wanted Bedspacer	Yes	No
Rico B.	Alab ng Puso	**Yes**	**Yes**
Rico B.	Basketbol	No	No
Rico B.	Isang Bandila	No	No
Rico B.	Kagat ng Lamok	Yes	No
Rico B.	Panahon Na Naman	No	No

In order to determine possible synergy between the two sets of feature described above, we combined them into a single data set and tested it using a multilayer perceptron with 25 hidden units and training time set to 70 iterations. The model was tested for 5 times using 10-fold cross-validations. As shown in Table 8, this resulted to an average kappa statistic of 0.72 and correctly classified instance of 81.02%.

Table 9 is the updated misclassification table after going through the results of the combined feature sets. As earlier hypothesized, the song lyrics that were marked as "Yes" for both Bigram and Trigram were no longer misclassified for the combined feature set. Some of the songs marked "Yes" for either Bigram or Trigram were also no longer misclassified for the combined feature set. However, the combined features set also introduced new misclassified lyrics. For instance, in the case of Overall, the number of misclassified lyrics was reduced. The number of misclassified Chito Miranda songs was reduced from 3 to 2 while the number of misclassified Ely Buendia songs was reduced from 13 to 7. Finally, the number of misclassified Rico Blanco songs remained the same at 5.

Table 8. Function Word Ranked Nominal + Bigrams + Trigrams Test Results

Test no.	Correct	Incorrect	Kappa Statistic
1	83.67%	16.33%	0.76
2	79.59%	20.41%	0.69
3	80.61%	19.39%	0.71
4	77.55%	22.45%	0.66
5	83.67%	16.33%	0.76
Average	**81.02%**	**18.98%**	**0.72**

Table 9. Updated Table of Misclassified Songs

Composer	Song Lyrics (FuncWord Only)	Song Lyrics (Func + Bigram + Trigram)	Ok with Bigram?	Ok with Trigram?
Chito M.	Sorry Na	-	**Yes**	**Yes**
Chito M.	The Crush	-	No	Yes
Chito M.	Wag Mo Na Sana	-	Yes	No
Chito M.	-	*Simbang Gabi*	-	-
Chito M.	-	*Trip*	-	-
Ely B.	Easy Ka Lang	-	Yes	No
Ely B.	Ang Huling El Bimbo	-	Yes	No
Ely B.	Harana	-	Yes	Yes
Ely B.	Kailan	Kailan	No	No
Ely B.	Kaliwete	-	No	Yes
Ely B.	Maskara	Maskara	No	Yes
Ely B.	Minsan	-	No	Yes
Ely B.	Overdrive	Overdrive	No	No
Ely B.	Pare Ko	Pare Ko	No	No
Ely B.	Sa Wakas	-	**Yes**	**Yes**
Ely B.	Spoliarium	Spoliarium	No	No
Ely B.	Walang Nagbago	-	**Yes**	**Yes**
Ely B.	Wanted Bedspacer	Wanted Bedspacer	Yes	No
Ely B.	-	*Kananete*	-	-
Rico B.	Alab ng Puso	-	**Yes**	**Yes**
Rico B.	Basketbol	Basketbol	No	No
Rico B.	Isang Bandila	-	No	No
Rico B.	Kagat ng Lamok	Kagat ng Lamok	Yes	No
Rico B.	Panahon Na Naman	Panahon Na Naman	No	No
Rico B.	-	*Para Hindi Ka Mawala*	-	-
Rico B.	-	*Himala*	-	-

Note: Song lyrics are considered consistently misclassified if they are incorrectly classified in at least 3 out of the 5 10-fold cross-validation tests.

Table 10 shows the confusion matrix from the final model with values averaged from the 5 tests and rounded to the nearest integer. Based on the confusion matrix, songs written by Chito Miranda appear to be the easiest for the model to properly identify while songs of Ely Buendia are the most difficult to properly attribute.

Table 10. Confusion Matrix (Average)

Actual Composer	***Predicted Composer***		
	ChitoM	ElyB	RicoB
Chito M	30	3	0
Ely B	2	23	7
Rico B	0	6	26

7. CONCLUSION

We have successfully implemented a machine learning method for automatically classifying song lyrics to its most likely composers using function words and character n-grams, two of the most commonly used textual features for authorship attribution. The techniques that we adopted are general enough to be applied in other domains that deal with short texts.

For the experiments conducted, we were constrained by the size of our data set to further improve or even disprove and test the limits of our chosen feature sets. Furthermore, we were not able to fully exploit song-specific features due to low performance results on initial tests conducted and our limited technical knowledge of song structures and songwriting in general. As a note on method, when pursuing the extraction and use of song-specific features it is important to gather data, as in this case song lyrics, that are officially published by the actual composer to avoid possible noise and bias introduced by the process of transcribing the song lyrics.

8. ACKNOWLEDGMENT

We are grateful to the Filipino Society of Composers, Authors, and Publishers (FILSCAP) for their assistance in identifying the songs written by the selected music composers.

9. REFERENCES

[1] Abbasi, A., and Chen, H. 2005. Applying authorship analysis to extremist group web forum messages. IEEE Intelligent Systems, 20(5), 67–75.

[2] Argamon, S., and Levitan, S. 2005. Measuring the usefulness of function words for authorship attribution. In Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing.

[3] Carletta, J. 1996. Assessing agreement on classification tasks: the kappa statistic. Computational linguistics, 22(2), 249-254.

[4] Chaski, C.E. 2005.Who's at the keyboard? Authorship attribution in digital evidence investigations. International Journal of Digital Evidence, 4(1).

[5] Fisette, M. 2010. Authorship identification in short text. Bachelor theses. Raboud Universiteit Nijmegen. http://www.nici.ru.nl/~idak/teaching/batheses/MarciaFissette_scriptie.pdf

[6] Frantzeskou, G., Stamatatos, E., Gritzalis, S., and Katsikas, S. 2006. Effective identification of source code authors using byte-level information. In Proceedings of the 28th International Conference on Software Engineering, 893–896.

[7] Grant, T.D. 2007. Quantifying evidence for forensic authorship analysis. International Journal of Speech Language and the Law, 14(1), 1–25.

[8] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutermann, P., and Witten, I.H. 2009. The WEKA Data Mining Software: An Update. SIGKDD Explorations, 11(1).

[9] Hoorn, J, Frank, S., Kowalczyk, W., and van der Ham, F. 1999. Neural network identification of poets using letter sequences. Lit Linguist Computing, 14(3), 311-338.

[10] Lat, J., Ng, S., Sze, K., Yu, G., and Lim, N. 2006. Lexicon acquisition for the English and Filipino language. 3rd National Language Processing Symposium. College of Computer Studies. De La Salle University - Manila.

[11] Ledger, G.R., and Merriam, T.V.N. 1994. Shakespeare, Fletcher, and the Two Noble Kinsmen. Literary and Linguistic Computing, 9, 235–248.

[12] Legara, E.F., Monterola, C., and Abundo, C. 2011. Ranking of predictor variables based on effect size criterion provides an accurate means of automatically classifying opinion column articles. Physica A, 390, 110-119.

[13] Stamatatos, E. 2009. A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology 60, 3, 538-556.