Automatic Lyrics Classification System Using Text Mining Technique

Chatklaw Jareanpon, Waranyoo Kiatjindarat, and Thanawat Polhome
Polar lab, Department of Computer Science,
Faculty of Informatics, Mahasarakham University
Mahasarakham, Thailand, 44150
Chatklaw.j@msu.ac.th, waranyoo.kia@msu.ac.th,
thanawat.pol@msu.ac.th

Kittiya Khongkraphan
Department of Mathematics and Computer Science,
Faculty of Science and Technology,
Prince of Songkla University,
Pattani Campus, Thailand, 94000
kittiya.k@psu.ac.th

***Abstract*—The human listens to the music for the entertainment and emotional expression. Nowadays, the music has many emotion, such as the love-music etc. The listener's playlist is made by manual. This paper proposes the automatic lyrics classification system using text mining technique by Naïve Bayes and Random forest based on the various weighting technique. The best of the accuracy is Naïve Bayes by the longest matching algorithm.**

Keywords—Text Mining, Classification, Lyrics

I. INTRODUCTION

The music is used for the human's emotional expression. It is one of the literatures, which the human understands easily. Normally, it is used for communicating. People are easy to remember, because of easy and short word. The difference between the music and the traditional communication are rhythm and melody, which only has in the music. Additionally, the word in music looks like poem, that it has the alliteration and assonance. The music has and represents many emotions, for example the love-music, the miss-music, and the heartbroken music etc. Nowadays, the playlist is consisted in many applications such as Youtube, iTunes etc. Playlist is a term to describe a list of video or audio files that can be played back on a media player sequentially or in random order. The playlist is usually made by manual. People sometime need to make the playlist by their emotion. This paper proposes the automatic lyrics classification system using text mining technique, especially in Thai language based on emotion.

II. LITERATURE REVIEW

Word segmentation is importance topic since segmented words are used in many applications such as speech synthesis and language translation applications. Several algorithms are proposed to Thai word segmentation that can be group into 3 main groups: rule based, dictionary based and text corpus based approaches. Mahatthanachai [1] reported that main effect of performance decreasing in word segmentation is not appeared of some technical term words or place name in dictionary. To deal unknown words, they proposed method to segment word using Parsing Thai Text with Syntax and Features of words (PTTSF). Hulth and el at. [2] proposed methodology for full text indexing, because they studied to select the importance word for searching in the document. Promjan and Teng-Amnuay [3] studied to compare the performance of the program and the algorithm of the word segmentation in Thai language. The best of the accuracy is the longest matching. The best of the recall is the back-track fixing algorithm. Furthermore, the structure of the dictionary is effective in the word segmentation. Chengzhi and el at. [4] studied the keywords for labeling in a document. They used the keyword extraction based on CRF. Although numerous approaches are proposed to deal Thai word segmentation problem, the performances of those methods are still much lower than the expectations. A main effect to performance decreasing of Thai word segmentation is combining of Thai and English language in one document. In this paper, we propose a novel approach using rule and dictionary based to increase accuracy of that word segmentation. This proposes approach outperforms several existing methods, especially in case of a document combined with that and English languages.

III. CLASSIFIER

The learning algorithm is separated into 2 types that are Eager and Lazy learning algorithm. They are different in term of model construction. This paper proposes to compare the accuracy between both by selecting the representative algorithms.

A. Naïve Bayes from Lazy learning algorithm

The concept of Naïve Bayes is based on probability, the theory in statistic. It is a supervised learning algorithm. In data mining, it is used for classification calculated by the equation (1)

$$P(C \mid B) = \underset{(C_j \in C)}{argMax}(P(C_i)) \prod_{(i=1)}^{n} P(B_j \mid C_i) \quad (1)$$

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

when C is the unknown data.

B is a word from the feature extraction process ($B = \{B_1, B_2, ..., B_n\}$).

B_j is an attribute at index j.

C_i is a class at index i.

B. *Random Forest or Ensemble model from Eager learning algorithm*

The Random forest is an ensemble learning algorithm. The algorithm steps are shown is Figure 4.

IV. METHODOLOGY

The methodology in this paper can be divided into 4 processes: 1) Data collection, 2) Data preprocessing, 3) Classification, and 4) Evaluation as shown in Figure 1.

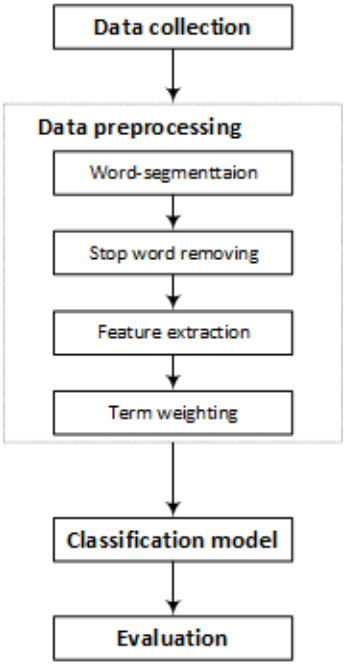

Fig. 1. Proposed methodology diagram

A. *Data collection*

The lyrics in Thai language are collected from the internet and labeled from thirty people. A training set has 120 lyrics, which can be divided into 3 groups of emotions: 1) 44 love lyrics, 2) 52 miss lyrics, and 3) 24 heartbroken lyrics. Furthermore, leave-p-out and K-fold cross validation are used to test the model.

B. *Data preprocessing*

In this process, it has 4 sub steps: 1) Word segmentation, 2) Stop word removing, 3) Feature extraction, and 4) Term weighting.

2.1 Word segmentation

This paper compares between the longest matching algorithm and the shortest matching algorithm, because the end of each Thai language word doesn't have the white space for the separating word, same as Chinese and Japanese. To selecting Thai dictionary, we use Lexitron [5].

2.1.1) The longest matching algorithm

It is a greedy algorithm, because it scans a message from left to right and selects the longest word, then finds in a dictionary as shown in Figure 2. For example, the result will show in Figure 5.

2.1.2) The shortest matching algorithm

Like the longest matching algorithm, the shortest matching algorithm is a greedy algorithm. It scans a message from right to left and selects the shortest word, then finds in a dictionary as shown in Figure 3. For example, the result will show in Figure 6.

2.2) Stop word removing

We remove the non-significance words, that usually are articles, conjunction, preposition, and pronoun. Because these words are always found, they affect to the performance.

2.3) Feature extraction

The objective of feature extraction is the word extraction that is founded in the document for creating the bag of words. This paper selects to use the 100 most frequently word.

C. *Term weighting*

Term weighting is normalization from word to weight vector. Several methods are used to calculate the weight vector such as Boolean weighting. This paper selects the 3 various methods that are Boolean, Term Frequency (TF) and Entropy weighting. They can calculate from equation (2)-(4).

1) Boolean weighting is calculated from

$$W_{ik} = \begin{cases} 1 \ if \ Tf_{ik} > 0 \\ 0 \ otherwise \end{cases} \tag{2}$$

2) Tf-weighting (Term frequency) is calcualted from

$$W_{ik} = Tf_{ik} \tag{3}$$

3) Entropy weighting is calculated from

$$entropy_{ik} = log\left(Tf_{ik} + 1.0\right) \times \left(1 + \frac{1}{log\left(N\right)} \sum_{j=1}^{M} \left[\frac{Tf_{ij}}{n_i} \times log\left(\frac{f_{ij}}{n_i}\right) \right] \right) \tag{4}$$

D. Classifier

We use the Naïve Bayes and Random Forest Algorithm and compare the accuracy between the different types of learning algorithms that are Eager and last learning algorithm.

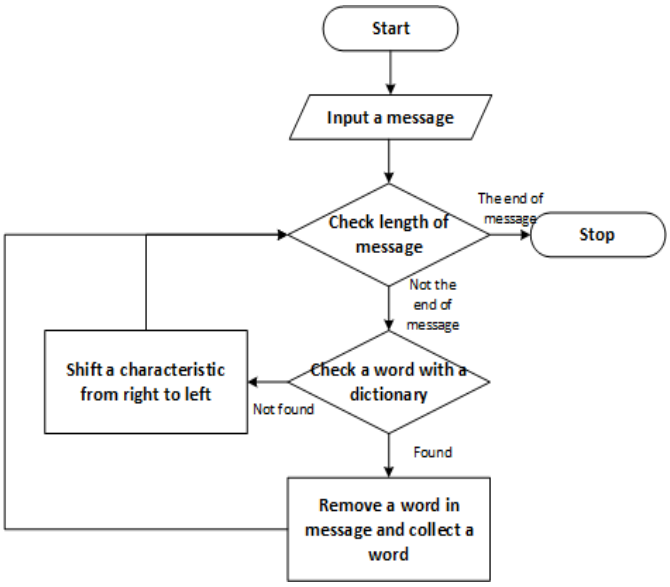

Fig. 2. The longest matching algorithm

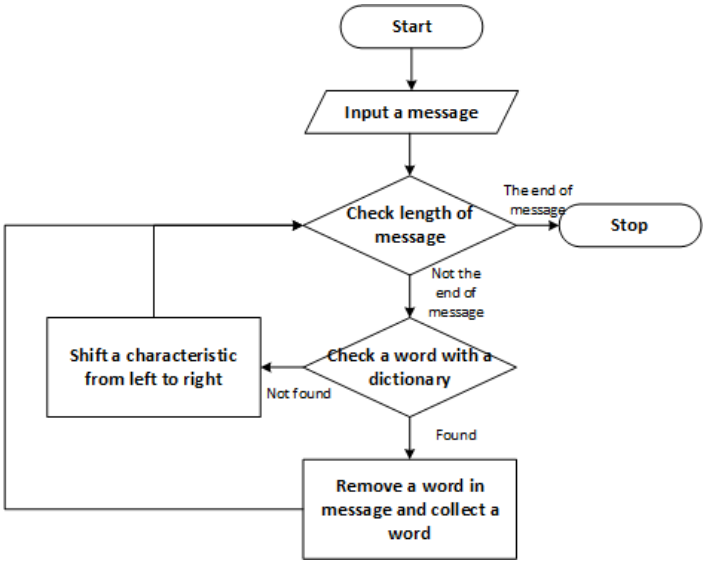

Fig. 3. The shortest matching algorithm

E. Evaluation

We evaluate our method in term of time and accuracy. The accuracy is calculated from Table I and equation (5)-(7).

TABLE I. ACCURACY VARIABLES

		Predicted Label	
		Positive	**Negative**
Known Label	Positive	TP	FN
	Negative	FP	TN

where

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$P_j = \frac{TP_j}{TP_j + FP_j} \quad (6)$$

$$R_j = \frac{TP_j}{TP_j + FN_j} \quad (7)$$

V. RESULT

The 120 lyrics are used in this experiment that is categorized into 3 groups: love, miss and heart-broken music. The average word of lyrics is 262.9 words. The accuracy and time of the word segmentation are shown in Table II.

This paper uses the leave-p-out cross validation and k-fold cross validation to test the model. For leave-p-out cross validation, the 80% of data are used to train the model and 20% of all classes are used to test the model. The K of k fold cross validation is set to 10. The 100 words are set the bag of word.

TABLE II. ACCURACY OF WORD SEGMENTATION

	Word segmentation	
	Longest	**Shortest**
Accuracy (1 lyrics 286 words)	86.03	15.38
Average time	0.358 second	0.957 second
Standard Deviation	0.171	0.254
Min (157 words)	0.056 Second	0.353 Seconds
Max (447 words)	1.168 Second	1.361 Second

This paper set the C1 is love, C2 is heartbroken and C3 is miss music. The accuracy, precision and recall are shown in Table III and IV, respectively.

VI. CONCLUSION AND FUTURE WORK

This paper proposes the methodology for automatic classification the lyrics using text mining technique. The best accuracy comes from Naive Bayes using Shortest matching algorithm. The miss-classification is consisted of various classes such as “heart”, “don’t known” that effect to the performance. These words will organize to stop word or feature selection. The future work of this research will add the context such as melody for adding the performance of classifier.

REFERENCES

[1] Mahatthanachai C., “Development of thai word segmentation technique for solving problems with unknown words”, 2015 International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2015.

[2] Anette H. “Automatic Keyword Extraction Using Domain Knowledge”, Journal of computational information system. 2008 .

[3] Promjan P. and Teng-Amnuay, “The analysis of performance comparison of Thai Word segmentation”, Faculty of Science, Chulalong korn University, 1997.

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features F, and number of trees in forest B.

```
function RandomForest(S, F)
    H ← ∅
    for i ∈ 1, ..., B do
        S^(i) ← A bootstrap sample from S
        h_i ← RandomizedTreeLearn(S^(i), F)
        H ← H ∪ {h_i}
    end for
    return H
end function
function RandomizedTreeLearn(S, F)
    At each node:
        f ← very small subset of F
        Split on best feature in f
    return The learned tree
end function
```

Fig. 4. Random Forest Classifier [6]

[4] Chengzhi Z. “Automatic Keyword Extraction from Documents Using Conditional Random Fields”, Journal of Computational Information Systems, 2008

[5] Lexitron Thai Dictionary, http://lexitron.nectec.or.th/.

[6] Matthew N. Bernstein, Note of Random Forest Algoritm, http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf.

ฉันกำลังนั่งตากลม - “ฉัน | กำลัง | นั่ง | ตาก | ลม”

Fig. 5. The longest matching algorithm result

ฉันกำลังนั่งตากลม - “ฉัน | กำ | ลัง | นั่ง | ตา | กลม”

Fig. 6. The shortest matching algorithm result

TABLE III. ACCURACY FROM LEAVE-P-OUT CROSS VALIDATION

Algorithm	Word Segmentation	Weight methods	Accuracy	precision			Recall		
				C1	C2	C3	C1	C2	C3
Naive Bayes	Longest Matching	Boolean	56.66%	0.36	0.6	0	0.4	0.9	0.4
		Term Frequency)TF(	**66.66%**	0	1	0.5	0	1	1
		Entropy	**66.66%**	0	1	0.5	0	1	1
Random Forest		Boolean	33.33%	0.25	0.36	0	0.3	0.7	0
		Term Frequency)TF(	33.33%	0	0.4	0	0	1	0
		Entropy	33.33%	0	0.4	0	0	1	0
Naive Bayes	Shorted Matching	Boolean	30%	0.28	1	1	0.7	0.2	0
		Term Frequency)TF(	33.33%	0	0	0.33	0	0	1
		Entropy	33.33%	0	0	0.33	0	0	1
Random Forest		Boolean	36.66%	0.5	0.52	0	0.1	1	0
		Term Frequency)TF(	33.33%	0	0.33	0	0	1	0
		Entropy	33.33%	0	0.33	0	0	1	0

TABLE IV. ACCURACY FROM 10-FOLD CROSS VALIDATION

Algorithm	Word Segmentation	Weight methods	Accuracy	STD
Naive Bayes	Longest Matching	Boolean	45.83%	8.84
		Term Frequency)TF(	19.99%	1.86
		Entropy	19.16%	2.28
Random Forest		Boolean	41.83%	6.72
		Term Frequency)TF(	18.33%	3.72
		Entropy	24.99%	11.78
Naive Bayes	Shorted Matching	Boolean	77.53%	20.99
		Term Frequency)TF(	70.99%	14.85
		Entropy	**77.29%**	**5.500**
Random Forest		Boolean	39.16%	11.25
		Term Frequency)TF(	40.16%	11.37
		Entropy	40.16%	11.37