Text Normalization on Thai Twitter Messages using IPA Similarity Algorithm

Sanphet Poolsukkho
Department of Computer Science,
Faculty of Science and Technology, Thammasat University,
Pathumthanee, Thailand
sanphet0638@gmail.com

Rachada Kongkachandra
Data Science and Innovation Program,
Faculty of Science and Technology, Thammasat University,
Pathumthanee, Thailand
rdk@sci.tu.ac.th

***Abstract*—Twitter often contains many noisy short messages. The noisy text are caused by insertion, transformation, transliteration and onomatopoeia. Text normalization is used for solving these noisy text. In this paper, we present the algorithm that can normalize insertion and homophonic transformation words by converting to International Phonetic Alphabet(IPA) and find the most similarity IPA of out-of-vocabulary and IPA of in-vocabulary using Levenshtein Distance. We used Twitter corpus that contained 2,000 twitter messages for evaluating the proposed algorithm. The experiment result illustrated that the proposed algorithm returned an accuracy of 79.03% when compared to dictionary-based normalization of LextoPlus returned an accuracy 24.19%.**

***Index Terms*—Text normalization; International Phonetic Alphabet; Twitter; Levenshtein Distance**

I. Introduction

Unknown words or out-of-vocabularies (OOV) are a major problem in natural language processing (NLP), especially text processing since they directly affect the processing performances in terms of accuracy and coverage. This problem is often originated by unintensional and intensional misspelling. An amount of OOV is highly depended on the type of text sources. The official text sources such as official document contains the small number of OOV, but text sources from social networks such as twitter may contain many OOV. The main reason is the post originators are free to post the text in their slang or individual style. The type of OOVs in social network thus is different from the other sources.

Twitter is one of the social media platforms that people used for showing their opinions, expressing their feelings, and communicating to others. The number of twitter's users is about 313 million users around the world [1] and therefore generally create 500 million tweets (twitter messages) per day [2]. The massive twitter message could be utilized in various types of analysis such as the community size analysis, keyword analysis, and sentiment analysis. However, the contents in tweets mostly contain many with out-of-vocabularies (OOV), which are words in different spelling but same in pronunciation.

In this paper, we focus on the OOVs having different spelling but same pronunciation. Tweet originators may choose to intentionally replace one alphabets or more from its proper word but to maintain or to similarlize the pronunciation of the word. For example, 'luv' may be used instead of 'love' and 'becoz' or 'bcoz' instead of the proper 'because'. Moreover, there are cases that the last alphabet of a word can be repeated to express their shouting gesture. These issues thus cause a low proficiency of the NLP system if they are not fixed.

In this work, we aim to normalize the above-mentioned OOV problem using the pronunciation. Hence, the unknown words will be transformed into international phonetic alphabet (IPA) [3]. Afterwards, the IPA of OOVs will be examined using similarity comparing to proper words in a dictionary. This work aims to study on Thai Tweeter; hence, the work applies the proposed method to Thai tweet. We expect that the normalization of these OOVs in tweeter should be beneficial to NLP tasks. The rest of this paper is structured as follows. In Section II, reviews on some related works in text normalization are given. In Section III, we present the process of IPA similarity algorithm. In Section IV, the evaluation result of the proposed method is described. Last, Section V gives a conclusion of the paper and the future work.

II. Related Work

There are many approaches in text normalization including Lexicon-based approach, Machine Learning-based approach and Hybrid Model-based approach. Lexicon-based approach is

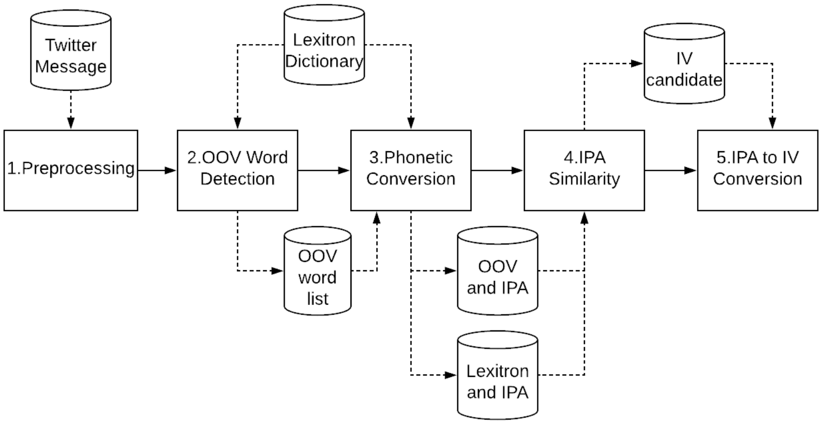

Figure 1. System Overview

a approach for normalizing OOV to in-vocabulary(IV) words by using rules which we create by hand. There are two popular models in Lexicon-based approach i.e. dictionary-based and phonetic-based model. Han et al. research [5] and Supranovich et al. research [6] use dictionary-based model for text normalization in twitter and then they get F1-score for 0.622 and 0.8337, respectively. The problem of dictionary-based model is when a new form of OOV word is coming, the rules have to be updated manually. Phonetic-based model approach is to convert OOV and IV words into phonetic string and then find the candidates having the most similar phonetic pattern. Ceron-Guzman et al. research [8] uses contextual information for finding candidate of OOV based on similarity of graphemes or phonemes and then they get F1-score for 69.53%. Khoury et al. research [9] uses radix tree for converting IV and OOV words to phonetic strings(IPA) to find the most probabilities similarity phonetic string with the overall accuracy result 80%.

Machine Learning-based approach is a approach for normalizing OOV to IV words by using machine-learning techniques. Rohatgi et al. research [7] uses deep learning for text normalization and get the overall accuracy 97.62%. The problem of this approach is it has to used many training data for converting OOV to IV words.

Hybrid-based approach is designed for normalizing OOV to IV words using two models or more. Satapathy et al. research [10] use two models i.e. dictionary-based and phonetic-based for doing the text normalization on twitter and get overall accuracy for 85.31%.

Haruechaiyasak et al. research [11] defines the intentional spelling errors into 4 categories i.e. insertion, transformation, transliteration and onomatopoeia. Insertion is the error that is occured whether user inserts one or more repeated character to the end of the word. Transformation is the error happen when the IV word has transformed into another form of word. There are two sub-types of transformation error i.e. homophonic and syllable trimming. Homophonic is the transformed word that has same or similar pronunciation of original word. Syllable trimming is the transformed word with remove one or more syllable from original word. Thai Transliterated words are created from another language and easily found in social media. Onomatopoeia words are created from Thai character set to emulate different sound in nature and environment including human and animal. LextoPlus is word segmentation and normalization tool that can normalize only insertion word form(remove repeated character).

III. Methodology

This paper aims for normalizing OOV words in the type of insertion and homophonic transformation on Thai twitter messages by using IPA similarity that is illustrated in figure 1. There are 5 main modules in this section i.e. preprocessing , OOV word detection , phonetic conversion , IPA similarity and IPA to IV conversion.

A. Preprocessing

In this module, we need to prepare twitter message by doing two sub-modules as follows.

1) Non-content Removal: The objective of this sub-module is to eliminate some characters that are not significant for detecting OOV word in twitter messages. For example, URL, emoji, emoticon, username, number, characters in other languages and special characters.

2) Word Segmentation: Since Thai sentence do not have an explicit word boundary such as a use of white space, we need this sub-module to segment Thai twitter messages into segmented word list by using Cutkum [12] which is Thai word segmentation framework using Long-Short Term Memory(LSTM).

B. OOV Word Detection

This module is designed to detect OOV words in a twitter message by doing sub-modules as follows.

1) IV-word Removal: There are IV and OOV words in segmented word list. This sub-module objective is to remove all segmented words in segmented word list that is matched with IV word from Lexitron dictionary [13] which is containing 35,328 Thai general standard words by NECTEC.

2) Collect OOV Words: This sub-module collects exist words from segmented word list to OOV word list.

C. Phonetic Conversion

Table I
THAI CONSONANT CHARACTERS AND IPA SYMBOLS PAIRS

SYMBOL	IPA		SYMBOL	IPA	
	INITIAL	FINAL		INITIAL	FINAL
ก	k	k$^{\urcorner}$	ท	t^{h}	t$^{\urcorner}$
ข	k^{h}	k$^{\urcorner}$	ธ	t^{h}	t$^{\urcorner}$
ฃ	k^{h}	k$^{\urcorner}$	น	n	n
ค	k^{h}	k$^{\urcorner}$	บ	b	p$^{\urcorner}$
ฅ	k^{h}	k$^{\urcorner}$	ป	p	p$^{\urcorner}$
ฆ	k^{h}	k$^{\urcorner}$	ผ	p^{h}	–
ง	ŋ	ŋ	ฝ	f	–
จ	tɕ	t$^{\urcorner}$	พ	p^{h}	p$^{\urcorner}$
ฉ	tɕ^{h}	–	ฟ	f	p$^{\urcorner}$
ช	tɕ^{h}	t$^{\urcorner}$	ภ	p^{h}	p$^{\urcorner}$
ซ	s	t$^{\urcorner}$	ม	m	m
ฌ	tɕ^{h}	–	ย	j	– or n
ญ	j	n	ร	r	n
ฎ	d	t$^{\urcorner}$	ล	l	n
ฏ	t	t$^{\urcorner}$	ว	w	–
ฐ	t^{h}	t$^{\urcorner}$	ศ	s	t$^{\urcorner}$
ฑ	t^{h}	t$^{\urcorner}$	ษ	s	t$^{\urcorner}$
ฒ	t^{h}	t$^{\urcorner}$	ส	s	t$^{\urcorner}$
ณ	n	n	ห	h	–
ด	d	t$^{\urcorner}$	ฬ	l	n
ต	t	t$^{\urcorner}$	อ		–
ถ	t^{h}	t$^{\urcorner}$	ฮ	h	–

In this module, we need to convert IV word from Lexitron and OOV word list to International Phonetic Alphabet (IPA) for finding the similarity between IPA of OOV word and IPA of IV word. Due to Thai language consists of 44 consonant characters, 21 vowel characters and 4 tone diacritics. Table I shows how to pair Thai consonant symbols to IPA symbol. Before converting to IPA symbols, Thai words have to seperate word element first. Then pair word with Thai consonant and vowel symbols to IPA symbols for converting to IPA string with text-to-phonetic rules which explain steps as follow.

1) Thai word is seperated as initial consonant, vowel and final consonant character(if final consonant character is consisted in Thai word).
2) Initial consonant, vowel and final consonant character is paired with IPA symbol.
3) IPA symbols string is sorted by IPA of initial consonant, IPA of vowel and IPA of final consonant character.

The difficulty of this task is the OOV words cannot convert to IPA because the new pattern of typing, for example, Thai word "ฮรืออ"(someone's sound crying). This word is inserted Thai consonant character to the middle of the word it is out-of-pattern that can convert to IPA.

We use Epitran [14], the python library, for converting OOV and IV word to IPA. The example is shown in table II.

Table II
EXAMPLE OF THAI WORDS CONVERSION TO IPA USING EPITRAN(SOME MAYBE INCORRECT)

Thai words	**IPA**
ขอโทษ	k^{h}ɔːt^{h}oːt
มาก	maːk
รัก	rak
สวัสดี	sawatdiː
อ้วน	ɔːwon

D. Find IPA Candidate

This module is designed to compare the difference between IPA of OOV word and IPA of IV word using Levenshtein Distance. Levenshtein Distance [4] is a edit distance for measuring the difference of two strings and shows how to calculate as follow Algorithm 1.

Algorithm 1 Levenstein Distance

```
function LEVENSTEIN($char\ seq1[1...m], char\ seq2[1...n]$)        ▷ Where seq1 - OOV word, seq2 - IV word
    $declare\ d[0...m, 0...n]$
    $set\ each\ element\ in\ d\ to\ zero$
    for $i\ from\ 1\ to\ m$ do
        $d[i, 0]\ =\ i$
    end for
    for $j\ from\ 1\ to\ n$ do
        $d[0, j]\ =\ j$
    end for
    for $j\ from\ 1\ to\ n$ do
        for $i\ from\ 1\ to\ m$ do
            if $seq1[i] = seq2[j]$ then
                $substitutionCost = 0$
            else
                $substitutionCost = 1$
            end if
            $d[i, j] = minimum(d[i-1, j]+1, d[i, j-1]+1, d[i-1, j-1]+substitutionCost)$
        end for
    end for
    $return\ d[m, n]$
end function
```

Table III
THE EXAMPLE OF CALCULATE LEVENSHTEIN DISTANCE BETWEEN IPA OF "ขอโทด" AND "ขอโทษ"

	IPA of "ขอโทด"	k	ʰ	ɔ	ː	t	ʰ	o	ː	t
IPA of "ขอโทษ"	0	1	2	3	4	5	6	7	8	9
k	1	0	1	2	3	4	5	6	7	8
ʰ	2	1	0	1	2	3	4	5	6	7
ɔ	3	2	1	0	1	2	3	4	5	6
ː	4	3	2	1	0	1	2	3	4	5
t	5	4	3	2	1	0	1	2	3	4
ʰ	6	5	4	3	2	1	0	1	2	3
o	7	6	5	4	3	2	1	0	1	2
ː	8	7	6	5	4	3	2	1	0	1
t	9	8	7	6	5	4	3	2	1	0

Table III shows how to calculate Levenshtein Distance between IPA of "ขอโทด" and "ขอโทษ". The score of this edit distance is 0 which mean the IPA of "ขอโทด" and the IPA of "ขอโทษ" is the same. So, we can use IPA of "ขอโทษ" to normalize.

E. IPA to IV Conversion

This module converts IPA of IV candidate word to replace each OOV word in the twitter messages.

IV. EXPERIMENTAL AND SETUP

In this paper, we manually collect 2,000 unique twitter messages dataset from TWEEPY API [15] which is the twitter message open free streaming api for collecting twitter message. Then, we detect OOV word from twitter messages and select only insertion and homophonic transformation words by hand. The number of interested intensional spelling error word is shown in table IV.

Table IV
THE NUMBER OF INTERESTED INTENSIONAL SPELLING ERROR WORDS

Type of interested intensional spelling errors	**Number of words**
Insertion	57
Homophonic Transformation	67
Total	**124**

After that we use OOV words for converting to IPA and find IPA candidate of IV word with Levenshtein Distance score 0. We evaluate the results using accuracy which is calculated as:

$$Accuracy = \frac{Number\ of\ correct\ results}{Number\ of\ total\ results} \tag{1}$$

The evaluation is summarized in table V. The accuracy of IPA similarity calculate by the number of correct result is the number of all the IPA of OOV word with Levenshtein Distance score 0 with IPA of IV word. The total result is the number of interested intensional spelling error words. The accuracy of IPA similarity is 79.03% when compare to the accuracy of dictionary-based (DCB) normalization (Norm) algorithm from LextoPlus is 24.19%. The result shows that when we solve more type of intensional spelling error, the higher of accuracy we will get.

Table V
EVALUATION RESULT

Algorithm	Accuracy
DCB Norm	24.19%
IPA Similarity	79.03%

V. DISCUSSION AND CONCLUSION

In this paper , we apply IPA similarity algorithm on Thai OOV words to do the text normalization on Thai twitter message. We conduct the experiment to find the best IPA candidate of OOV words by using Levenshtein Distance. The accuracy of proposed algorithm is 79.03%. We show that IPA similarity algorithm is better than DCB Norm algorithm. The experiment verify that solving the insertion and homophonic transformation word is significant for improving text normalization on twitter.

Due to the OOV words we detect, sometime it is showing that homophonic transformation word type does not has the same pronounce, for example, "อ้วน" is changed to "ต้วน" or "อ้วง" in English is mean "fat". The first or last letter has changed so the Levenshtein Distance score cannot give the zero score. More problem is OOV word has combination error type, for instance, the word "ฮรืออออ" is from "ฮือ" in English mean the sound of human cry. This word is inserted alphabet to the middle and end of the word so the Levenshtein Distance cannot give score 0 too.

For future work, we would like to improve the performance of proposed algorithm by adding more rule about the changed pronounce word.

REFERENCES

[1] Number of monthly active twitter users worldwide from 1st quarter 2010 to 3rd quarter 2016 (in millions). [Online]. Available: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[2] Twitter usage statistics. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/

[3] "International_Phonetic_Alphabet @ en.wikipedia.org." [Online]. Available: https://en.wikipedia.org/wiki/International{_}Phonetic{_}Alphabet{#}cite{_}note-IPA{_}1999-1

[4] G. Navarro, "A Guided Tour to Approximate String Matching 1 Introduction," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.

[5] B. Han and T. Baldwin, "Lexical Normalisation of Short Text Messages : Makn Sens a # twitter," *Computational Linguistics*, vol. V, no. 212, pp. 368–378, 2011. [Online]. Available: http://www.aclweb.org/anthology/P11-1038

[6] D. Supranovich and V. Patsepnia, "IHS_RD: Lexical Normalization for English Tweets," no. 2011, pp. 78–81, 2015. [Online]. Available: https://noisy-text.github.io/2015/pdf/WNUT11.pdf

[7] M. Zare and S. Rohatgi, "DeepNorm-A Deep Learning Approach to Text Normalization," 2017. [Online]. Available: http://arxiv.org/abs/1712.06994

[8] J. A. Cerón-Guzmán and E. León-Guzmán, "Lexical Normalization of Spanish Tweets," *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, pp. 605–610, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2872518.2890558

[9] R. Khoury, "Microtext normalization using probably-phonetically-similar word discovery," *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2015*, pp. 384–391, 2015.

[10] R. Satapathy, C. Guerreiro, I. Chaturvedi, and E. Cambria, "Phonetic-Based Microtext Normalization for Twitter Sentiment Analysis," *IEEE International Conference on Data Mining Workshops, ICDMW*, vol. 2017-November, pp. 407–413, 2017.

[11] C. Haruechaiyasak and A. Kongthon, "Lextoplus: A thai lexeme tokenization and normalization tool," *WSSANLP-2013*, p. 9, 2013.

[12] P. Treeratpituk, "Thai Word-Segmentation with Deep Learning in Tensorflow." [Online]. Available: https://github.com/pucktada/cutkum

[13] "Dictionary Thai-English named LEXiTRON version 2009." [Online]. Available: http://lexitron.nectec.or.th/2009{_}1/

[14] D. R. Mortensen, S. Dalmia, and P. Littell, "Epitran: Precision G2P for many languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. Paris, France: European Language Resources Association (ELRA), May 2018.

[15] "Tweepy:An easy-to-use Python library for accessing the Twitter API." [Online]. Available: http://www.tweepy.org/