Code Switching Phenomenon in Thai Tweets

Ben P. Athiwaratkun
Statistical Science Department
Cornell University
pa338@cornell.edu

***Abstract*—Code switching is a phenomenon where there is a change in the language in a given phrase or sentence. This paper aims to investigate the Thai-English code-switching phenomenon that occurs in Thai tweets. We analyzed the distribution of code-switching English words compared to the word distribution for normal English usage. We also analyzed the Thai 1,2,3-grams that are correlated with the occurrence of code-switching with a ridge regression model. This model also gives us a predictor of the occurrence of code-switching which performs slightly better than a flat predictor.**

I. Introduction

A. Motivation

Below are examples of code-switching tweets involving the word `organize`.

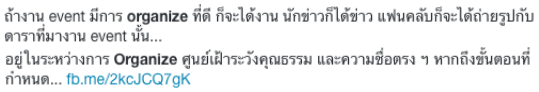


Figure 1: Example of Code-Switching Tweets

The reasoning for choosing the word `organize` instead of an alternative word in the base language is not readily clear. Both of these tweets mainly talk about event organization which does not specifically require the English word. The hypotheses are that the choice to use English reflects the socio-economic status. It can also be the case that the word is easier to say than the Thai counterpart. Or the word might help convey the meaning better. Which hypothesis is true is a particularly difficult question. Instead, this paper will investigate the structures of the code-switching instances. In particular, the questions we will answers are the following:

- What English words occur significantly more or less in code-switching instances compared to its normal English usage? Is there any clustering structure we can infer from these words?
- What Thai words or phrases are significantly correlated with the occurrences of code-switch? Can we use these words to predict whether the author will code-switch in a Tweet?

B. Data Choice

Code-switching can be observed in many forms of communication methods, particularly in informal ones such as posts on social media platforms and forums. `Twitter` is a particularly suitable platform due to its convenient API. Note that the general methodology presented in this paper can also be extended for bigger dataset that contains not only Thai but other languages. However, for this paper, the dataset is limited to only tweets in Thai as data cleaning process heavily requires knowledge in the base language.

C. Related Work

To add

II. High Level Summary

Two major parts of this paper are:

(i) Analysis of code-switching words

- We look at the part of speech of code-switching English phrases and code-switching English unigrams as opposed to their normal English usage counterparts.
- We filter out and select only English unigrams that are non-proper nouns as true code-switching instances and compare them with normal English distribution, both individually and cluster-based.

(ii) Code-switching prediction

- We perform logistic regression with target variable being an indicator $(0, 1)$ of whether a code-switching happens in a tweet and the features being the occurrences of n-grams.

III. Methodology

This section presents a data preparation process for both (i) and (ii). The steps in shaded boxes are analyses of results which are explained in Section IV.

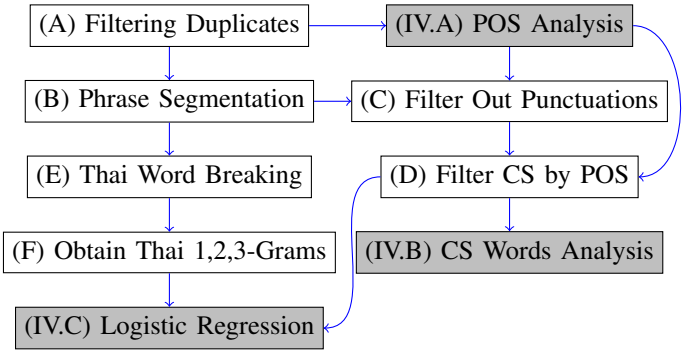


Figure 2: Data Pre-Processing

A. Filtering Duplicates

This phase ensures that there are no duplicate tweets in the data set. The data used are streaming data which contains very little duplicate tweets (same id), however, we find 55 tweets with duplicate `id_str` out of $2,583,847$ tweets.

Note that the analysis of code-switching words without filtering retweets out is equivalent to treating a retweet as a regular tweet, which could be justifiable if the non code-switching tweets are as likely to be retweeted as the code-switching tweets. However, this assumption might not hold. Additionally, to perform regression, we need to filter out retweets in order to eliminate the possibility of prediction of seen data in the test set. We filter out tweets that contain the following regular expression pattern `r'RT @[\w]+:'` This filter is needed in addition to the `"retweeted":true` tag contained in tweet json format as manual inspection confirms that some tweets that are clearly retweets have this flag set as `"retweeted":false`. The final dataset contains $775,400$ tweets

B. Phrase Segmentation

C. Word Breaking (Thai)

For a given Thai phrase or sentence, words are usually written contiguously without any space between them. In addition, spaces sometimes are also omitted between English word and Thai words as shown below.

Note: This Tweet has been abridged and modified for demonstration purpose.

First, we process tweets by selecting out contiguous Thai characters by regular expression pattern `u'[\u0E00-\u0E7F]+'` which matches Thai characters.

For each thai phrase, we use the library `libthai0.1.4` by Theppitak Karoonboonyanan et al. and a Python interface `PyThai` to break each Thai phrase into a list of words. For example, the example tweet becomes

D. Obtaining N-Grams

Figure **??** shows the histogram of Thai $1, 2, 3$-grams obtained from sliding window method on the lists of contiguous Thai words. That each, we treat each input, e.g., the following three lists as disjoint.

Note that we treat any white space, punctuation, and English character as a stopping character for the n-gram sliding window. For instance the segmented list `[???]` is not included in any $2-$ or $3-$ gram, which should be the case particularly for Thai because the space in the example tweet indicates the end a phrase.

FIGURE of histogram

E. Design Matrix and Labels

mention sparse

Explain regression

We consider only n-grams that occur at least 10 times as features and build a sparse matrix of size $NUMTWEET \times NUMDIM$ as a design matrix for regression.

F. Part of Speech Analysis

G. Punctuation Filtering

H. CS Labeling

Based on the analysis in section (F), we decide to filter out words with part of speech not in the following list.

TABLE with explanation

I. CS Words Analysis

J. Regression Label

IV. Results

A. Part of Speech of Code-Switching Words

We define pseudo code-switching instances as the occurrences of English phrases that are adjacent to Thai phrases. These are not all what we consider true code-switching instances since words such as 'Harry Potter' or 'Interstellar' almost, if not always, necessitates the use of the English words.

The hypothesis is that these pseudo code-switching instances will contain a large number of proper nouns. In addition to proper noun category (denoted by ^), we also found marked differences in other part of speech's probability, as shown in Figure 3 and 5. The POS legend and examples are in Table I

Table I: Part of Speech Legend and Examples. See **??** for more details.

POS Tag	Tag Meaning	Examples
N	common noun	`mv line BTS mama cap winner`
O	pronoun	`me I Me II US i`
^	proper noun	`iPhone ELF Facebook 2NE1`
L	nominal + verbal	`ibaekrauhls ID iPhone6Plus`
V	Verb	`vs talk VS read VOTE Rewind`
A	Adjective	`infinite Favorite Fast`
R	Adverb	`Y forever here alone always`
!	Interjection	`gt http lt amp IG`
P	subordinate conjunction	`via by in Like`
&	coordinating conjunction	`n and or But`
T	verb particle	`up off down out`
#	hashtag	`#codeswitching`
@	at-mention	`@user`
~	discourse marker	`RT rt Rt PLSRT -rt`
U	URL or email address	`google.com/dtJfg04`
E	emoticon	`x XD T___T T__T -w-`
$	numeral	`2ndWin One 1stWin 10thirty`
,	punctuation	`yg_bear YG_iKONph i5 -v-`
G	other abbreviation	`SM iKON w M PlsRT`

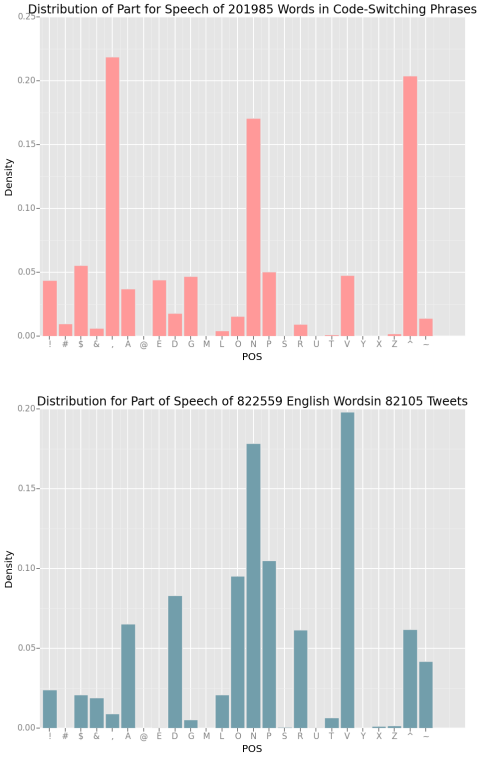


Figure 3: Distribution of Part of Speech for Words in Code Switching Phrases vs Words in English Tweets

Figure 3 shows that estimate of the probability for each part of speech (MLE estimate in this case, which is equivalent to the observed ratio with no smoothing) of words that occur in pseudo code-switching English phrases, as well as the part of speech for English words in normal English tweets.

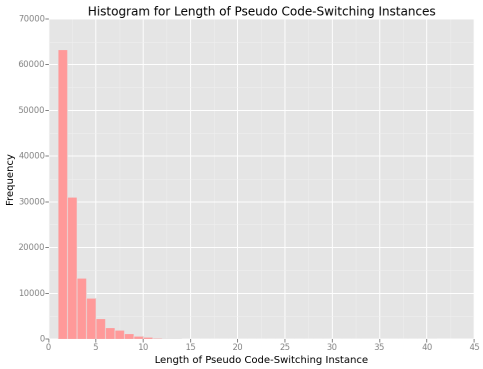


Figure 4: Histogram for number of words in pseudo code-switching phrases. There are $775,345$ pseudo code-switching phrases contained in $102,550$ Tweets.

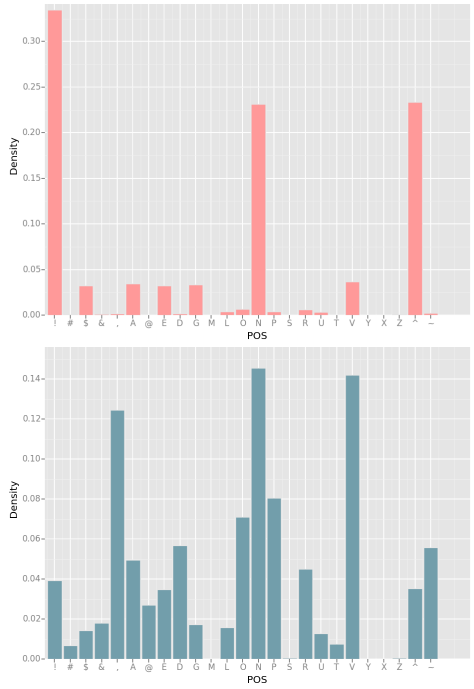


Figure 5: Distribution of Part of Speech for Words in Code Switching Unigrams vs Words in English Tweets

From this point on, we will turn our attention to code-switching words that are meaningful English words and are not proper nouns. From the preliminary analysis presented, we decide to keep only the words with the following part of speech: `[&,A,O,N,P,R,T,V]` Note that ideally, we should be able to keep a larger set part of speech. However, based on the tags performed on a subsets of twitter words shows in Table I, we choose to filter out some potentially meaningful part of speech that ideally we should be able to use such as L (nominal + verbal).

In addition, there are some proper nouns that passed through the part of speech filter such as `interstellar, divergent, vine, galaxy`. This group contains proper nouns that might be recently popular and consequently is not incorporated in the POS tagger. We manually go through the code-switching words (this is possible due to its relatively small size of distinct words) to identify proper nouns and filter out words in the following list:

B. Code-Switching Unigrams

Figure 6 shows the words and their associated code-switching probabilities and English tweet probabilities. The probabilities here refer to the MLE probability estimates with (no smoothing) with the group of $2,673$ distinct words that occur in both code-switching and English usage. The total number of code-switching words are $13,175$ and the total number of English words are $518,410,750$.

On average, each word occurs roughly $\frac{13,175}{2,673} \approx 5$ times in code-switching. Figure 6 show the words that occur in code-switching at least 10 times.

Table II: List of Proper Nouns for Manual Filter

Proper Noun	Explanation
'interstellar'	Movie name
'divergent'	Movie name
'insurgent'	Movie name
'kamikaze'	Music band
'line'	App name
'marvel'	Company name
'vine'	App name
'whiplash'	Movie name
'beam'	Singer name
'coke'	Company/Product name
'muggins'	Harry Potter term
'tot'	Company name
'galaxy'	Product name

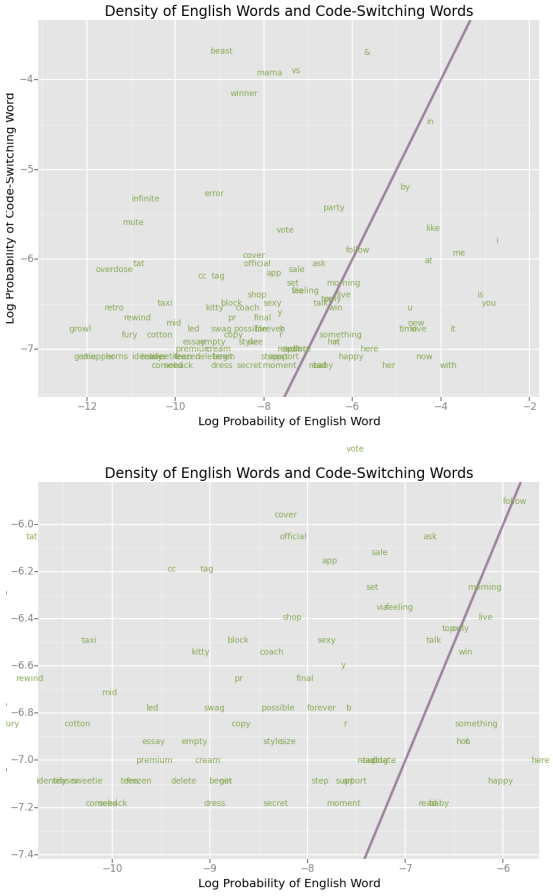
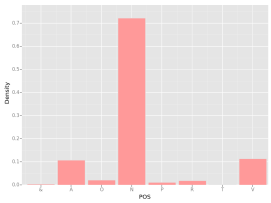
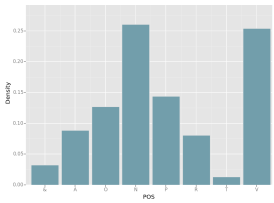


Figure 6: Distribution of part of speech for words in code switching unigrams versus words in English tweets. The second plot is the original plot but shows only the lower middle region of the original plot.

We consider the distribution of part of speech again but only for code-switching unigrams after filtering process. We also compare the code-switching word with the English unigrams filtered with the part of speech set `[&,A,O,N,P,R,T,V]` . The English dataset are words occurred in tweets from 2008 to 2012. This is obtained from `Brown Cluster Data` Figure 5 shows the result.



(a) Distribution of POS for Code-Switching Words



(b) Distribution of POS for English Words

Figure 7: Distribution of Part of Speech for Words in Code Switching Unigrams vs Words in English Tweets

Next, we attempt to compare the code-switching words with English words using Brown cluster algorithm.

show plot

C. Logistic Regression

We perform a regularized logistic regression with 5-fold cross-validation to determine the optimal regularization parameter λ. The penalty term considered are both ℓ_1 and ℓ_2 norm of the coefficient vector $\vec{\beta}$. Logistic regression with very high dimensional feature space (order of magnitude $10^5 - 10^6$) is quite efficient compared to other classifiers. We use the library `scikitlearn` with the design matrix X in `Python`'s `scipy` sparse matrix format.

Table IV shows the optimal parameter $\lambda = 1.$ based on the percentage of correctness score. We also analyzes the prediction power of this logistic regression as shown in Table **??**.

Table III: Average of prediction scores of 5-fold cross validations for both lasso and ridge regression. The predication score is defined by the percentage of correct prediction.

Regularization	λ	Prediction Score	Standard Deviation
Lasso	e^{-4}	0.972047	3.6637×10^{-4}
Lasso	e^{-3}	0.976628	3.6681×10^{-4}
Lasso	e^{-2}	0.980829	3.6053×10^{-4}
Lasso	e^{-1}	0.984475	1.5892×10^{-4}
Lasso	e^{0}	0.986524	7.2210×10^{-5}
Lasso	$\mathbf{e^{1}}$	**0.986922**	$\mathbf{9.77314 \times 10^{-5}}$
Lasso	e^{2}	0.986816	2.15595×10^{-5}
Lasso	e^{3}	0.986745	3.14490×10^{-5}
Lasso	e^{4}	0.986494	7.93141×10^{-6}
Ridge	e^{-4}	0.979649	3.08827×10^{-4}
Ridge	e^{-3}	0.982186	2.71405×10^{-4}
Ridge	e^{-2}	0.984274	1.56487×10^{-4}
Ridge	e^{-1}	0.985726	1.23270×10^{-4}
Ridge	e^{0}	0.986445	9.32293×10^{-5}
Ridge	e^{1}	0.986806	4.93549×10^{-5}
Ridge	$\mathbf{e^{2}}$	**0.986917**	$\mathbf{4.79543 \times 10^{-5}}$
Ridge	e^{3}	0.986903	3.95688×10^{-5}
Ridge	e^{4}	0.986524	2.18813×10^{-5}

With this optimal λ obtained from cross validation, we perform logistic regression on the entire dataset and analyze the significant features.

D. Significant Thai N-Grams

Table V lists Thai N-grams with positive betas that are significant at level $\alpha = 0.001$.

Table IV: Prediction scores for the baseline estimator (predict all non code-switch) and the classified trained by regularized logistic regression.

Estimator	λ	Accuracy	Precision	Recall
Baseline Estimator	N/A	0.0	0.025499	
Trained Lasso Logistic Regression	e^2	2.0	0.062459	

Table V: List of Thai $1, 2, 3$-Gram with Postive Betas

N-Gram	Translation	β
Column-wise Constant α	1.0	0.025499
Column-wise Constant α	2.0	0.062459
Column-wise Constant α	4.0	0.130297
Column-wise Constant α	8.0	0.302554
Column-wise Constant α	16.0	0.751134
Column-wise Constant α	32.0	2.415786
Element-wise Constant α	1.0	0.000389
Element-wise Constant α	2.0	0.001880
Element-wise Constant α	4.0	0.006716
Element-wise Constant α	8.0	0.003288
Element-wise Constant α	16.0	0.006320
Element-wise Constant α	32.0	0.022578

Next, we demonstrate sample code-switching tweets that contain at least one the significant n-gram with positive beta.

V. Discussion

This section discusses the challenges the further improvement that can be made.

A. Code-Switching Determination

One of difficult tasks in this research is the determination of whether a code-switch occurs in given tweet. The process employed filters out proper noun quite well to some extent. However, we find that there are some words that the POS tagger could not have known. An example of this is the following tweet.

Figure 8: Example Tweet

The word `Line` in this case refers to a messaging application that is particularly popular in South East Asia. In this paper, we manually goes through the words and select out the set of words that are most likely proper nouns such as `line, interstellar`, to name but a few. However, it could very well be the case that the tweet actually refers to the word `line` in the usual English meaning. An improvement for the code-switching determination process would be a classifier that translates the whole phrase into English and performs the tagging.

B. Sparsity of Code Switching Data

In this exercise, the code-switching tweets occurs with frequency roughly 2%. This translates to about 13K code-switching instances with 2.6K distinct words. This is quite small compared to the number of words in the English baseline of 800K distinct words. Given more time and resources, this project can be extended to a larger data set. If we gather enough data such that the number of distinct code-switching words is close to 800K, this will allow for more rigorous analysis on the comparison of code-switching and English word clusters.

VI. Conclusion

The conclusion goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] K. Gimple, N. Schneider, B. O'Conner et at., *Part-of-Speech Tagger for Twitter: Annotation, Features, and Experiments*

[3] P. Liang. *Semi-Supervised Learning for Natural Language*, Department of Electrical Engineering and Computer Science, MIT, 2005. `https://github.com/percyliang/brown-cluster`