Pattern Matching Refinements to Dictionary-Based Code-Switching Point Detection

Nathaniel Oco
De La Salle University
2401 Taft Avenue Malate, Manila City
1004 Metro Manila, Philippines
nathan.oco@delasalle.ph

Rachel Edita Roxas
De La Salle University
2401 Taft Avenue Malate, Manila City
1004 Metro Manila, Philippines
rachel.roxas@delasalle.ph

Abstract

This study presents the development and evaluation of pattern matching refinements (PMRs) to automatic code switching point (CSP) detection. With all PMRs, evaluation showed an accuracy of 94.51%. This is an improvement to reported accuracy rates of dictionary-based approaches, which are in the range of 75.22%-76.26% (Yeong and Tan, 2010). In our experiments, a 100-sentence Tagalog-English corpus was used as test bed. Analyses showed that the dictionary-based approach using part-of-speech checking yielded an accuracy of 79.76% only, and two notable linguistic phenomena, (1) intra-word code-switching and (2) common words, were shown to have caused the low accuracy. The devised PMRs, namely: (1) common word exclusion, (2) common word identification, and (3) common n-gram pruning address this and showed improved accuracy. The work can be extended using audio files and machine learning with larger language resources.

1 Introduction

Code-switching (CS) is "the use of two or more linguistic varieties in the same interaction or conversation" (Myers-Scotton and Ury, 1977). It is often prevalent in communities where there is language contact. According to linguistic studies (Bautista, 1991; Bautista, 2004; Borlongan, 2009), code-switching reasons are mainly driven by proficiency or deficiency in the languages involved. Proficiency-driven code-switching takes place when a person is competent with the two languages and can easily switch from one to the other "for maximum efficiency or effect". On the other hand, deficiency-driven code-switching takes place when people are forced to code-switch to one language because they are "not competent in the use of the other language". Oral communication in both languages can be enhanced by the detection of code-switching points (CSPs). To detect CSPs, we developed a dictionary-based approach using a rule-based engine (Naber, 2003), and we also developed pattern matching refinements (PMRs) to improve accuracy.

As testbed, this study focuses on Tagalog-English code-switching, which can be classified into (1) intra-sentential and (2) intra-word code-switching. Intra-sentential CS is the switching between Tagalog and English words and clauses, while intra-word CS is the use of English root words with Tagalog affixes and morphological rules. An example of intra-sentential CS is "Unless let us say ***may*** *mga bisita siya*" (translated as: Unless let us say he/she has visitors) and an example of intra-word CS is "*nagdadrive*" (incompeleted aspect of the English verb "drive"). The system developed can effectively be used to detect intra-sentential (Tagalog to English and English to Tagalog) and intra-word CSPs.

26th Pacific Asia Conference on Language,Information and Computation pages 229–236

This paper is organized as follows: related works in section 2, CSP detection in section 3, pattern matching refinements in section 4, testing and discussion in section 5, and conclusion in section 6.

2 Related Works

In the field of computing, several studies have been done to automatically detect CSPs. The areas that are commonly involved are machine learning, audio signal processing, and natural language processing (NLP). In machine learning, patterns are derived from large data sets such as in the CSP studies of Spanish-English (Solorio and Liu, 2008) and Chinese-English (Burgmer, 2009), which used the transcription of forty minutes and four hours of audio recordings, respectively. In audio signal processing, analyses of speech corpora (e.g. the Cantonese CUSENT and the English TIMIT) using acoustic models (White et al., 2008) are studied. Analyses of trained phone models (Chan et al., 2004) are also studied.

In NLP, a related study (Yeong and Tan, 2010) explored n-gram-based approaches and also presented dictionary-based approaches. N-gram-based approaches such as alphabet bigram, grapheme bigram, and syllable structure use similarity measures and language models extracted from a corpus. On the other hand, dictionary-based approaches such as language vocabulary list and affixation information match the word against a dictionary. Table 1 shows a performance comparison of different NLP approaches (Yeong and Tan, 2010). The table shows that dictionary-based approaches yield lower accuracy rates than model-based approaches and are known to have lower performance ratings.

Approach		Accuracy
Dictionary -based	Affixation Information	76.26%
	Vocabulary List	75.22%
N-gram-based	Alphabet Bigram	91.29%
	Grapheme Bigram	91.82%
	Syllable Structure	93.73%

Table 1: Performance comparison of different NLP approaches (Yeong and Tan, 2010)

3 CSP Detection

The system has been plugged into OpenOffice and it highlights CSPs in an OpenOffice document. Figure 1 shows a sample screenshot of the system detecting CSP in the sentence "And then ***kinuha niya***" (translated as: And then he/she took it).

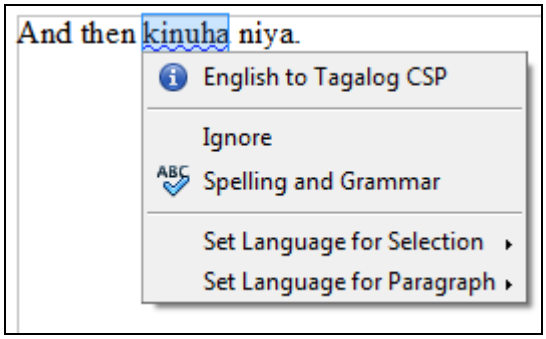

Figure 1: Sample screenshot of the system showing English to Tagalog CSP

After studying the Philippine component of the International Corpus of English (Bautista et al., 2004), we experimented on a dictionary-based approach to detect CSPs using LanguageTool (Naber, 2003) – a rule-based style and grammar checker engine that can run as an OpenOffice extension. Figure 2 shows the architecture of the developed system. LanguageTool requires two language resources to run: (1) the tagger dictionary and (2) the rule file. For the tagger dictionary, we utilized and edited word declarations from the English (ENG TD) and Tagalog (TAG TD) supports. For the rule file (RF), we developed pattern matching rules.

CSP detection works as follows: (1) an input text document is separated into sentences and each sentence is separated into tokens; (2) each token is given their tag – English, Tagalog, Proper Noun, Punctuation, or UNKNOWN – using the tagger dictionary declarations; (3) the tokens together with their tags are matched against the rule file, which contains code-switching patterns that we declared; (4) if a pattern matches, the user is notified. In a related work (Oco and Borra, 2011), LanguageTool was used in Tagalog grammar checking. Thus, this study is the first attempt to use LanguageTool in another language processing task.

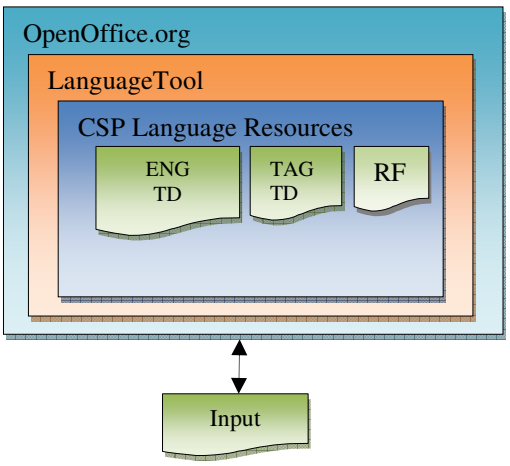

Figure 2: Architecture of the system

3.1 Tagger Dictionary

The CSP tagger dictionary contains approximately 298,498 words from the English language support, 7,441 words from Tagalog, 35 punctuation marks, 54,157 proper nouns and 1000 new word declarations, for a total of 361,131 words. File size was almost 10MB. We reduced it to 1MB by encoding[1] it to a smaller file, making it easier to load. Figure 3 shows sample word declarations. To distinguish between Tagalog and English words, a header tag was assigned to Tagalog words and a different one to English words. The words in the tagger dictionary are classified into four header tags: Tagalog words ("TAG"), English words ("ENG"), proper nouns ("NPRO"), and punctuations ("PSNS").

kasimputi	kasimputi	TAG ADCO S
#	#	PSNS
$	$	PSNS
nonsecluded	nonsecluded	ENG JJ
nonsecludedness	nonsecludedness	ENG NN
nonsecludedly	nonsecludedly	ENG RB
Abbottson	Abbottson	NPRO
Abboud	Abboud	NPRO
Abby	Abby	NPRO

Figure 3: Sample word declarations

LanguageTool supports different languages. As comparison, Table 2 shows the word count in six language supports. The numbers highlight that Tagalog is a poorly-resourced language.

Language Support	Word Count
German	4,158,968
Polish	3,662,366
French	550,814
English	354,744
Asturian	157,747
Tagalog	7,484

Table 2: Number of word declarations in six language supports

3.2 Rule File

The rule file, like any xml file, is composed of elements and attributes. Figure 4 shows a sample rule file. The three main elements are: (1) pattern, (2) message, and (3) example. Pattern refers to the token or sequence of tokens and/or part-of-speech (POS) to be matched; message refers to the feedback, which will be shown to the user if the pattern matches the input; and example refers to the sentences used for testing. If a pattern matches, CSPs are marked and message is shown to the user.

```
<rule id="ENGLISH-TAGALOG" name="Code
Switch to Tagalog">
<pattern case_sensitive="no" mark_from="1">
<token postag="ENG.*" postag_regexp="yes"/>
<token postag="TAG.*" postag_regexp="yes"/>
</pattern>
<message>English to Tagalog CSP</message>
<example type="incorrect">I want to be
<marker>sundalo</marker>.</example>
<example type="correct">They are
soldiers.</example>
</rule>
```

Figure 4: An English to Tagalog CSP rule using POS checking

Pattern matching in CSP detection works by checking if a word is English or Tagalog, i.e. has a header tag "ENG" or "TAG". This would result in false positives if it is a common word – a word that appears in both the English and Tagalog tagger dictionaries – as this would have both header tags. An example of a common word is "may" (e.g.

[1]Morfologik was used to convert text files to FSA-encoded .dict files. Morfologik is available at: http://sourceforge.net/projects/morfologik/files/

"may ENG…" and "may TAG…"), which could be an English Verb or a Tagalog existential marker. Another example is "raw", which could be an English adjective or a Tagalog enclitic.

Using POS checking, one true positive and one false positive are detected in the sentence "Unless let us say ***may*** *mga bisita siya*" (translated as: Unless let us say he/she has visitors). Both "*may*" and "*mga*" are detected as English to Tagalog CSP. We developed pattern matching refinements to improve accuracy.

4 Pattern Matching Refinements

Pattern matching refinements (PMRs) work by separating pattern matching for sentences involving common words and words with unknown POS. Figure 5 shows the diagram of the different word types. Words (W) are generally classified into four: unique English words (UEW), unique Tagalog words (UTW), common words (CW), and unknown words (W-(UEW ∪ UTW)). Unique English words are words with "ENG" header tags only. The same applies for Tagalog words ("TAG"). Since the English tagger dictionary is well-resourced, unknown POS indicate either intra-word code-switching or undeclared Tagalog words.

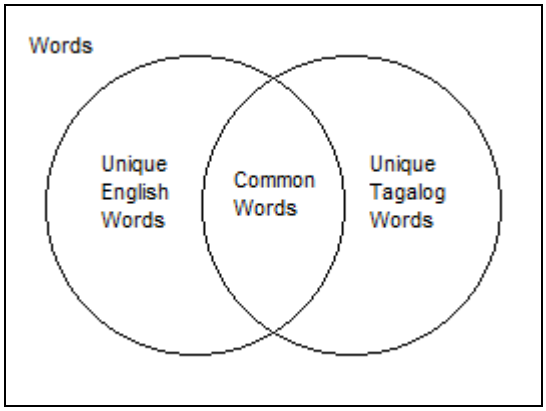

Figure 5: The set of words showing the intersection of UEW and UTW as common words

4.1 Common Word Exclusion

We developed common word exclusion to reduce the detection of false positives in sentences involving common words. The uniqueness of a word in a tagger dictionary is taken into consideration (i.e. a word does not have multiple declarations with different header tags) and common words are excluded from the pattern. If a unique English word is followed by a unique Tagalog word, then the second word is a CSP. The same applies if a unique Tagalog word is followed by a unique English word. Figure 6 shows a pattern without common word exclusion and Figure 7 shows a pattern with common word exclusion. The list of common words was generated by getting the intersection of word declarations with header tag "ENG" and those with header tag "TAG". For scalability, a new tag "CW" was created and common words were added in the tagger dictionary with this tag. This PMR is similar to common word pruning (Dimalen and Roxas, 2007), which was used as a language model improvement to increase the accuracy rate of language identification involving closely-related languages. The difference is common words are completely discarded in common word pruning while in this PMR, common words are excluded from the pattern and declared only as exceptions.

```
<token postag="ENG.*"
postag_regexp="yes"></token>
<token postag="TAG.*"
postag_regexp="yes"></token>
```

Figure 6: Pattern matching without common word exclusion

```
<token postag="ENG.*" postag_regexp="yes">
<exception postag="CW.*"
postag_regexp="yes">
</exception></token>
<token postag="TAG.*" postag_regexp="yes">
<exception postag="CW.*"
postag_regexp="yes">
</exception></token>
```

Figure 7: Pattern matching with common word exclusion

Using common word exclusion, no true positive nor false positive is detected in the sentence "Unless let us say ***may*** *mga bisita siya*" (translated as: Unless let us say he/she has visitors).

4.2 Common Word Identification

Since common words are excluded from pattern matching, common words that are also code-switching points are also not detected. We developed common word identification to identify the language of the word. This approach works by identifying which tokens or POS of tokens normally precede or succeed common words. Word window is one-previous and one-next. To determine word sequences, a word bigram model of English Wikipedia articles[2] was generated and bigrams involving common words were manually analyzed. POS sequences were derived and declared in the rule file. For example, if a common word has a verb tag and is preceded by an English verb, the common word is a CSP.

Using common word exclusion and common word identification, one true positive, "*may*", is detected in the sentence "Unless let us say ***may*** *mga bisita siya*" (translated as: Unless let us say he/she has visitors).

4.3 Common n-gram Pruning

The previous refinements do not detect words that are not declared in the tagger dictionary, i.e. words with UNKNOWN POS tag. We developed common n-gram pruning for this purpose. An n-gram is defined as an "n-character slice of a longer string" (Dimalen and Roxas, 2007). N-grams that are unique to a particular language are used and declared in the rule file. For example, if a word has an unknown POS tag and it contains n-gram sequences that are unique to English, then it is intra-word code-switching. To get the unique n-grams, n-gram profiles of the languages were generated using Apache Nutch[3]. A sampling of the English Wikipedia and the entire Tagalog Wikipedia[4] – containing approximately 10 million and 3 million words, respectively – were used as training data. Each generated n-gram profile contains approximately 500 bigrams, 3,000 trigrams, and 3,000 four-grams. Less than 50 unique n-grams were taken per language and regular expression was used for scalability. This PMR is similar to n-gram-based approaches (Yeong and Tan, 2010). However, in this study, no similarity measure was used, the number of characters varies in length, and all character sequences are unique to the language model. In a similar study, common word pruning (Dimalen and Roxas, 2007) was introduced to get the unique words. In this study, unique character sequences were instead generated.

[2] The English wiki articles are available in XML file format at this website: http://dumps.wikimedia.org/enwiki/

[3] http://nutch.apache.org/

[4] Tagalog wiki: http://dumps.wikimedia.org/tlwiki/

5 Testing and Discussion

Approximately one hour of audio recording of actual conversations was transcribed for the study. It contains more than 500 sentences and approximately 80% of these sentences contain CSPs. The first 100 sentences with CSPs were taken from the transcription and used to test the system. Audio recordings of actual conversations were used because they show the natural usage of the languages. The test corpus contains 820 words, 243 of which are CSPs and verified by an expert. Five separate tests were conducted: (T1) an initial test with POS checking only; (T2) with common word exclusion only; (T3) with both common word exclusion and identification; (T4) with common n-gram pruning only; (T5) with all pattern matching refinements – common word exclusion, common word identification, and common n-gram pruning. Table 3 shows the results. The number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) are indicated in third, fourth, fifth, and sixth column, respectively.

Test Type	PMR	Results			
		TP	FP	TN	FN
T1	None – POS checking only	166	89	488	77
T2	With common word exclusion only	158	6	571	85
T3	With common word exclusion and CWID	193	6	571	50

T4	With common n-gram pruning only	12	0	577	231
T5	With all PMRs	205	6	571	38

Table 3: Test results using 100 sentences containing 243 CSPs

With POS checking only, the system properly detected 166 CSPs but it also detected 89 instances of false positives. On the other hand, common word exclusion reduced the number of false negatives from 89 instances to 6. However, it does not detect common words that are also code-switching points, which is why the number of true positives is lower. When used with common word identification, the number of true positives increased to 193 instances. Meanwhile, common n-gram pruning detected 12 instances of intra-word CS that were not previously detected. Table 4 shows a list of properly detected verbs with intra-word CS. Syllable reduplication in contemplated and incompleted verb aspects was observed (e.g. *mag-aapprove*, *nagfoforum*). The number of true positives is low because common n-gram pruning detects only intra-word CS and words with unknown POS. A combination of all PMRs brings the total number of true positives to 205.

Token	Root Word	Aspect
idefault	default	Neutral
ikiclick	click	Contemplated
ipaupload	upload	Contemplated
mag-aapprove	approve	Contemplated
magmemorize	memorize	Neutral
mag-upload	upload	Neutral
nagclick	click	Completed
nagfoforum	forum	Incompleted

Table 4: Sample list of verbs with detected intra-word CS

Table 5 shows the accuracy, which increases as more PMRs are used. CSP detection using no pattern matching refinements yielded 79.76% accuracy. A comparison between our results and the results of a related work (Yeong and Tan, 2010) shows that basic dictionary-based approaches are not highly effective. Analyses of our results show that two phenomena cause false positives and false negatives. These are (1) intra-word CS and (2) common words – especially those with different semantic information. Consider the sentence "Unless let us say ***may*** *mga bisita siya*" (translated as: Unless let us say he/she has visitors). The word "*may*" is a common word. If a basic dictionary-based approach is applied, both "*may*" and "*mga*" are detected as CSPs. Table 6 shows a list of identified common words with different semantic information. Difference in the POS was observed.

Test Type	PMR used	Accuracy
T1	None – POS checking only	79.76%
T2	With common word exclusion only	88.90%
T3	With common word exclusion and CWID	93.05%
T4	With common n-gram pruning only	71.83%
T5	With all PMRs	94.51%

Table 5: Accuracy rate of the different tests conducted

Token	English POS	Tagalog POS
akin	Adjective	Pronoun
along	Preposition	Noun
at	Preposition	Conjunction
ate	Verb	Noun
away	Adverb	Verb
dating	Verb	Adjective
gusto	Noun	Verb
halos	Noun	Adjective
hanging	Verb	Noun
ho	Interjection	Polite Marker
kilos	Noun	Verb
may	Auxiliary Verb	Existential Marker
naming	Verb	Pronoun

Token	English POS	Tagalog POS
noon	Noun	Pronoun
paring	Verb	Noun
piling	Verb	Noun
raw	Adjective	Enclitic
ring	Noun	Enclitic
sawing	Verb	Adjective
tinging	Verb	Adjective

Table 6: Sample list of common words with different semantic information

A combination of all pattern matching refinements yielded the highest accuracy with 94.51%. This can be attributed to the detection of common words and intra-word CS.

A close analysis of the false negatives reveals that some intra-word CS was not detected. Table 7 shows a sample list. The n-gram sequence of these words is not unique and is found in the Tagalog n-gram profile. Intra-word CS with n-gram sequence similar to Tagalog words is not properly detected by the system.

Token	Root Word	Aspect
naghahang	hang	Incompleted
ilogin	login	Neutral
magregister	register	Neutral
inedit	edit	Completed
dinisable	disable	Completed
malilink	link	Contemplative
inonote	note	Contemplative
linalog	log	Incompleted
magreport	report	Neutral

Table 7: Sample list of verbs with intra-word CS that were not detected

6 Conclusion and Recommendation

This paper has shown that the application of PMRs significantly increased accuracy. The tests show that with all PMRs, the system was able to achieve 94.51% accuracy. The result is higher than no improvements used. The results are also higher than the results yielded by dictionary-based approaches in a related study (Yeong and Tan, 2010).

As future work, other forms of multilingualism can be considered. There are instances where more than two languages are involved in code-switching and these are rarely documented. Also, code-switching involving dialectal variations may be considered and since Tagalog is a poorly-resourced language, bootstrapping can be applied. Additional resources may also be added and machine learning be used, such as in (Solorio and Liu, 2008) and (Burgmer, 2009). Also, the work can be extended to cover audio files, such as in (White et al., 2008) and (Chan et al., 2004).

Acknowledgments

This work has been funded by the Department of Science and Technology - Philippine Council for Industry, Energy and Emerging Technology Research and Development (DOST-PCIEERD) through the "Inter-Disciplinary Signal Processing for Pinoys (ISIP): ICT for Education" Program and also supported by the University Research Coordination Office of De La Salle University (No. 20FU211).

We thank Prof. Dr. Ma. Lourdes S. Bautista, Dr. Ariane Borlongan, and Ms. Mary Anne Conde for being instrumental to the completion of this study. We also thank reviewers for their comments.

References

Ariane M. Borlongan. 2009. Tagalog-English Code-Switching in English Classes in the Philippines: Frequency and Forms. TESOL Journal, 1: 28-42.

Carol Myers-Scotton and William Ury. 1977. Bilingual Strategies: The Social Functions of Code-Switching. International Journal of the Sociology of Language, 13: 5-20.

Christoph Burgmer. 2009. Detecting Code-Switch Events based on Textual Features. Diploma Thesis. Karlsruhe Institute of Technology, Karlsruhe.

Christopher M. White, Sanjeev Khudanpur, James K. Baker. 2008. An Investigation of Acoustic Models for Multilingual Code-Switching. In Proceedings of the 9th Annual Conference of the International Speech Communication Association. Brisbane, Australia: International Speech Communication Association.

Daniel Naber. 2003. A Rule-based Style and Grammar Checker. Diploma Thesis. Bielefeld University, Bielefeld.

Davis Muhajereen D. Dimalen and Rachel Edita O. Roxas. 2007. AutoCor: A Query Based Automatic Acquisition of Corpora of Closely-related Languages. In Proceedings of the 21st Pacific Asia Conference on Language, Information, and Computation. Seoul, Korea: Korean Society for Language Information.

Joyce Y.C. Chan, P.S. Ching, Tan Lee, and Helen M. Meng. 2004. Detection of Language Boundary in Code-Switching Utterances using Bi-Phone Probabilities. In Proceedings of the 4th International Symposium on Chinese Spoken Language Processing. Hong Kong, China: Chinese University of Hong Kong.

Ma. Lourdes S. Bautista. 1991. Code-Switching Studies in the Philippines. International Journal of the Sociology of Language, 88: 19-32.

Ma. Lourdes S. Bautista. 2004. Tagalog-English Code-Switching as a Mode of Discourse. Asia Pacific Educational Review, 5 (2): 226-233.

Ma. Lourdes S. Bautista, Loy V. Lising, and Danilo T. Dayag. 2004. ICE-Philippines Lexical Corpus - CD-ROM. London: International Corpus of English.

Nathaniel A. Oco and Allan B. Borra. 2011. A Grammar Checker for Tagalog using LanguageTool. In Proceedings of the 9th Workshop on Asian Language Resources Collocated with IJCNLP 2011. Chiang Mai, Thailand: Asian Federation of Natural Language Processing.

Thamar Solorio and Yang Liu. 2008. Learning to Predict Code-Switching Points. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Honolulu, HI: Association for Computational Linguistics.

Yin-Lai Yeong and Tien-Ping Tan. 2010. Language Identification of Code-Switching Malay-English Words Using Syllable Structure Information. In Proceedings of the 2nd Workshop on Spoken Languages Technologies for Under-Resourced Languages. Penang, Malaysia: Universiti Sains Malaysia.