



A Text Clustering Preprocessing Technique for Mixed *Bisaya* and English Short Message Service (SMS) Messages for Higher Education Institutions (HEIs) Enrolment-Related Inquiries

Michelle Bao-Torayno^{1,*}, Love Jhoye M. Raboy² and
Consortio S. Namoco Jr.²

¹Quantumlinx PTY LTD, Sydney, NSW, Australia

²University of Science and Technology of Southern Philippines, Cagayan de Oro City, Philippines



Article Type: Article

Article Citation: Michelle Bao-Torayno, Love Jhoye M. Raboy, Consortio S. Namoco Jr. A text clustering preprocessing technique for mixed *Bisaya* and English short message service (SMS) messages for higher education institutions (HEI) enrolment-related inquiries. *Indian Journal of Science and Technology*. 2020; 13(06),654-673.DOI:10.17485/ijst/2020/v013i06/149363

Received date: December 2, 2019

Accepted date: January 17, 2020

***Author for correspondence:**

Michelle Bao-Torayno 
michellelorayno@yahoo.com 
Quantumlinx PTY LTD, Sydney, NSW, Australia

Abstract

Objectives: This study is aimed to develop a text preprocessing technique for mixed *Bisaya* and English short message service (SMS) messages. This technique is used to extract significant keywords for SMS message clustering procedure as the basis for SMS automated response on Higher Education Institution (HEI)'s enrollment-related inquiries. **Methods/statistical analysis:** In this study, a text clustering preprocessing technique is introduced and developed for mixed *Bisaya* and English SMS messages for Higher Education Institution (HEI) enrollment-related inquiries. The technique is a relatively new approach to extract significant keywords while addressing key challenges in morphological complexities on mixed *Bisaya* and English SMS messages. The method has seven (7) stages namely: tokenization, language tagging, stop-word removal, stemming, Soundex, final-tagging, and language translation. The term frequency co-occurrence clustering approach is applied to evaluate the precision and effectiveness of the text preprocessing technique. **Findings:** Test results revealed that the method produces a good preprocessing procedure with approximately 73%–83% accuracy rate on text processing and 87%–90% accuracy rate when text preprocessing is applied to clustering. **Application/improvements:** The results of this study may assist academic institutions in maximizing the opportunity to effectively entertain more enrollment-related inquiries via SMS as an alternative communication medium to its target market. This also promotes technological advancement for the institution as it utilizes an ICT-enhanced marketing approach through mobile technology.

Keywords: Text Preprocessing, Text Clustering, SMS Messaging, Stemming Algorithm, Enrollment-related Inquiries.

1. Introduction

Document or text clustering is an unsupervised classification of text collections into distinct groups of similar documents where similarity is defined as some function on documents. Generally, a text clustering algorithm partitions a document based on their topic similarities. This means that documents which discuss the same topic are assigned to a single cluster [1].

Recent developments on the Internet and mobile technologies resulted in an overwhelming growth of multilingual documents on the web and short messaging service (SMS) messages. These documents are written in numerous different languages and on diverse topics, and organizing these documents have become a critical problem. Due to the need for methods that deal with text collections in various languages simultaneously, there is also an increased demand for a robust multilingual document clustering algorithms.

Documents usually written in multilingual and informal style are short-length documents such as micro blog posts and SMS messages [2]. SMS is a means of sending short text messages not longer than 160 characters (including spaces) between mobile phone devices. The terms SMS is often used as a synonym for all types of short text messaging as well as the activity of creating SMS texts [3].

In the Philippines, SMS messages are usually written with morphological complexity and varied formats using a combination of English words and words from local dialects. In Cagayan de Oro City, in particular, *Bisaya* (a local dialect) and English terms are simultaneously used to send SMS messages. Aside from these, locals also use terms and acronyms which are only known in the locality to fit the 160-character limit (spaces and symbols included) for each message sent from mobile phone devices. As a result, preprocessing or parsing can be difficult for SMS messages in the locality due to the following key challenges:

- a) SMS messages are in free text form.
- b) SMS messages are morphologically complex.
- c) Local SMS messages also include slang words (e.g. lol for laughing out loud) and out of vocabulary words (e.g. *w8t* for wait, *niaq* or *nia q* for *nia ko* or I'm here) termed locally as *jejemon*.
- d) Words may have omitted letters or abbreviations (e.g. *pla* for *pila* <how much>, *nrol* <enroll>, *nrolmnt* <enrollment>, *dp* <down payment>, *btw* <by the way>, *tnx* <thanks>)
- e) SMS message includes “noises” such as spelling errors and incomplete sentences.

Several parsing algorithms have been formulated as preprocessing technique to deal with multilingual text documents. However, such algorithms are not suitable for SMS messages which are short-length documents [4]. Use of these existing parsing algorithms may lead to wrong extraction of relevant keywords which then results in an ineffective SMS clustering. Hence, it is necessary to propose and evaluate new text preprocessing techniques [5]. In Ref. [6], a system for bilingual lexicon extraction of terms from English and Tagalog comparable corpora is introduced. However, the system still has several issues that may affect the accuracy of extraction of words. Moreover, local dialects usually differ

from Tagalog in form and structure. One distinguishing feature of the *Bisaya* and Tagalog language is the difference and complexity of the prefix, infix, and suffix morphemes present in the two languages [7–8].

To overcome the shortcomings of the preprocessing techniques for short messages, and at the same time provide a suitable approach for the *Bisaya* dialect, this study developed a text preprocessing technique for mixed *Bisaya* and English SMS messages. This technique is used to extract significant keywords for SMS message clustering procedure as the basis for SMS automated response on Higher Education Institution (HEI)'s enrollment-related inquiries.

The enrollment-related inquiries through SMS, refer to questions or clarifications coming from interested parents, guardians and/or students about HEI's enrollment information such as tuition fees, entrance examination schedules, down payments, courses offered, etc. via SMS messages. These SMS messages are an alternative communication medium between HEI and its target market when internet is unavailable to access the website or phone call loads are insufficient.

This article is organized as follows. Section 1 provides an introduction to the study. A brief literature review on existing preprocessing techniques is presented. Section 2 describes the methods used in this study. The design and method used for the parsing algorithm, its rules and conditions, implementation and performance evaluation used in this study are also discussed in this section. Results and discussions are provided in section 3. A brief conclusion is provided in section 4.

2. Literature Survey

Several parsing algorithms have been formulated as a preprocessing technique to deal with multilingual text documents. However, these algorithms are not suitable for SMS messages which are short-length documents. The use of these existing parsing algorithms may lead to wrong extraction of relevant keywords and thus may result in an ineffective SMS clustering. Hence, it is necessary to propose and evaluate new text preprocessing techniques.

Preprocessing ensures that data are cleaned from missing values and inconsistencies. This step smoothens noisy data which are then used as inputs to the next step called clustering. The preprocessing step is crucial in determining the quality of the next stage, which is the classification or clustering stage.

In preprocessing, it is essential to select the significant keywords that carry the meaning of the document being examined and discard the words that do not contribute to distinguishing between the documents [9]. This may result in a positive or negative influence on its overall accuracy. Therefore, the right choice of the preprocessing approaches will lead, by necessity; to the improvement of any text mining tasks vary greatly. Moreover, it has already been proven that the time spent on preprocessing can take from 50% up to 80% of the entire classification process [10–11] which proves the importance of preprocessing in text classification process.

Different studies designed and implemented the variety of preprocessing steps due to the associated constraints that must be satisfied and fulfilled first with their application

and even its given languages. For example, web search result clustering with a problem on search performance utilizes a suffix tree clustering technique with preprocessing steps such as tokenization, stop-word removal, and stemming algorithm [12].

2.1. Tokenization

In computational linguistics, tokenization refers to the process of splitting a piece of text into a list of tokens. A token can be a word, a number, a symbol, or a punctuation mark. A most common and straightforward tokenization rule called “*black and white token*” is stated as, “*Split the character sequence at whitespace positions and cut off punctuation marks, parentheses and quotes at both ends of the fragments to obtain the sequence of tokens*”. This simple rule is quite accurate because whitespace and punctuation are reasonably reliable indicators of word boundaries. However, it is investigated [13] that some punctuation marks like period, apostrophe, and dash, treated as delimiters can cause disambiguation problem. Hence, the primary challenge lies in the proper treatment of symbols, digits, and punctuation marks.

The study of Han [14] implemented the so-called “*Improved Greedy Tokenizer*”. This approach does not allow the characters like apostrophe, period, and dash to join the separator or delimiter group; instead, these characters are handled case by case. This can be done by incorporating an additional procedure in the algorithm which either throws away or retains the symbols, digits, and punctuation marks in a token, as described in a sample algorithm shown in Figure 1.

On the other hand, the study of Khan et al. [15] performed the following procedures: (1) Punctuation removal; (2) Symbol removal; (3) Numeral removal; and (4) Transformation to lower-case. These procedures are obtained from third-party software called *AutoMap*. This software is a British–English stand-alone application that automates preprocessing text modification such as removal of symbols, lowercase conversion, bigrams, and name entity extraction, keywords in context, delete list, stemming, and generalization. *Auto Map* also has a support tool for the generation and editing of thesauri, ontologies and delete lists exist.

The importance of tokenization stems from the fact that the output from this process determines the keywords for the document collections to be used in the subsequent processes. The tokenizer function thus identifies those keywords and returns it to the stemming process.

```

If ( token Length > 1)

    If token starts and ends with " ' " then

        Remove starting and ending " ' "

    else if token starts with " ' " then

        Remove starting " ' "
  
```

FIGURE 1. Enhancement of an improved greedy algorithm.

2.2. Stop Word Removal

The **stop word algorithm** is implemented to remove meaningless words frequently occurring in the document. The removed words are then collected to form the *stop list*. Pronouns, particles, and prepositions (except of) are not extracted during parsing since these parts of speech (POS) are usually considered to be redundant in text mining. Additional concepts that are redundant are populated in the exclude list.

This algorithm is implemented in all documents preprocessing techniques since it eliminates words that are meaningless and does not affect the over-all clustering process in any way. In the English language, there are about 400–500 stop words. Examples of such words include “the”, “of”, “and”, “to”. The first step during preprocessing is to remove these stop-words, which has proven to be very important.

2.3. Stemming Algorithm

Krovetz stemmer which produces complete words as stems rather than truncated ones that can be produced by other stemmers. The Krovetzstemmer effectively and accurately removes inflectional suffixes in three steps: the conversion of plural to its single form (e.g. “-ies”, “-es”, “-s”), the conversion of past to present tense (e.g. “-ed”), and the removal of “-ing”. The conversion process firstly removes the suffix, and then through a process of checking in a dictionary for any recoding (also being aware of exceptions to the standard recoding rules), returns the stem to a word. The low level of strength with the English language due to the nature of the stemmer causes issues with its usage within the field of Information Retrieval (IR), where an increased level of strength and index compression may be sought. For this reason, this Stemmer is frequently used in conjunction with other stemmers, making good use of the advantage of the accuracy of removing suffixes by this Stemmer, which then adds the compression of another Stemmer, such as the Paice/Husk Stemmer or Porter Stemmer.

A stemming approach for Arabic word documents before document clustering. Arabic word stemming is a technique that aims to find the lexical root or stem forwards in natural language by removing affixes attached to its root. This is necessary because Arabic word can have a more complicated form with those affixes. Arabic stemming algorithms can be classified, according to the desired level of analysis, as *root-based* approach and *stem-based* approach. The root-based approach uses morphological analysis to extract the root of a given Arabic word. Many algorithms have been developed for this approach. Stem-based approach or Light Stemmer approach is not meant to produce the root of a given Arabic word. Instead, it is employed to remove the most frequent suffixes and prefixes. In addition, the term-document using Term Frequency-Inverse Document Frequency (TFIDF) weighting scheme is computed.

In Ref [2], the preprocessing activities play a vital role in various applications. Therefore, it is concluded that the domain specific applications are more proper for text mining. Their paper presented three essential preprocessing techniques, namely, stop word removal, stemming, and indexing. It also examined the different stemming algorithms that use table lookup by merely looking for the stem of a word in a table. Since such data are not readily

available and might require considerable storage space, this type of stemming algorithm might not be practical. Successor variety stemming is based on the determination of morpheme boundaries, uses knowledge from structural linguistics, and more complex than affix removal stemming algorithm.

In Ref. [16], various preprocessing techniques for unstructured data mining and its applications were surveyed. Stemming algorithms such as MF Porter and Krovetz were analyzed for their efficiencies and accuracies. Some of the drawbacks of Porter's algorithm are that it leads to a significant degree, it is context dependent and, it results in a wrong stem. The major drawback of Krovetz stemming algorithm is that it becomes inefficient with large input documents and its inability to deal with words that are not present in the lexicon.

A system using the correlation formula in [2] algorithm complemented with other components such as a part of speech tagger and a stemmer to extract bi-lingual terms from English and Filipino comparable corpora has five main components namely Preprocessor, Co-occurrence Analyzer, Computation of Similarity/Correlation, Selection of highly Similar/Correlated words, and the Lexicon Editor. However, the system still has several issues that may affect the accuracy of extraction of words. Moreover, local dialects usually differ from Tagalog in form and structure. One distinguishing feature of the *Bisaya* and Tagalog language is the difference and complexity of the affix morphemes present in the two languages.

TagSA, a Tagalog Stemming Algorithm was developed for all forms of Tagalog words. It is a new approach in stemming words having complex morphological structure such as Tagalog words. It does not only cover removal of prefixes, suffixes, infixes, and circumfuses but reduplication and compounding as well.

The general overview of the stemming process in TagSA is shown in Figure 2. The process starts when an input word is fed to the stemmer. The word then passes through 8 stage routines of the stemmer. Routines 1.0-2.0 refer to the non-stemming stage that handles the hyphen-search and dictionary-search routines. In this step, every removal of an affix requires a dictionary lookup to avoid overdoing a stemming process. An added procedure called the swapping inquiry takes place within suffix removal routine which

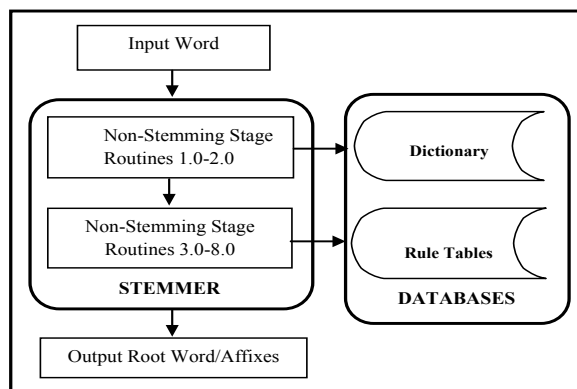


FIGURE 2. Architectural design of TagSA.

causes the original word to be recalled and reprocessed in the stemming stage with the prefix and suffix routines “swapped”.

TagSA is a dictionary-based algorithm that must contain a comprehensive list of Filipino root words. The handling of assimilatory and phoneme change rules was also added to resolve the complexity of words in the Tagalog language once fed into a machine translation system which includes semantic interpretation of the words used in a particular sentence or phrase.

2.4. Other Applied Techniques

Part of Speech (POS) tagging is an intermediate step that is carried out to identify concepts for language translation purposes. As an example, a POS tagger is incorporated twice in the process using a lookup table based on its lexicons. The first POS tagging takes place after the text file has been loaded while the second or final POS tagging is executed after the stemming process. It searches the lexicon repetitively for the words in the corpora, starting from the first word to the last, and returns a result each time. If the result returns exactly one match, then the part of speech tag attached to the word (in the lexicon) is assigned to the word. If, however, the word does not exist or that it has more than one part of speech tag, “no pos” is assigned.

The results of the experiment for this study show that having a database lookup as POS tagger does not decrease processing time, but instead causes the processing to take longer. Not only does it increase the total time of extraction, but it also does not improve accuracy. This is because the POS tagger’s processes are executed in a large text file and a more complex morphological feature of a Tagalog language that includes reduplication and compounding words. In addition, the words tagged will always be the known words from the database, while the untagged will still be the unknown words, not found in the database.

A method which uses a dictionary lookup to translate all documents in Japanese and Russian to English where no word sense disambiguation is performed. A machine translation system is used to translate documents in languages other than Russian and Japanese to English. The reason behind the usage of dictionaries for Japanese and Russian and machine translation for other languages is that they have created a fast technique for dictionary lookup in Japanese and Russian, but not for other languages. After translation is performed, the method proceeds by considering all documents as written in a uniform language and applies a general monolingual algorithm to discover document clusters. A simple procedure is done here as a proof-of-concept that full translation does not need to be performed for clustering.

TagSA can also be applied to other Philippine-type languages exhibiting the same structure as Tagalog by implementing some changes in rule tables and modifications in some sections of the algorithm and the dictionary. It should be noted, however, that the Bisayan language exhibits some characteristics different from that of Tagalog. The two differ in vocabulary, phonology, morphology, and syntax. Hence, there is a need to introduce a method that can be used for Bisayan.

The algorithms as mentioned earlier can produce quality solutions but cannot guarantee that they can produce an optimal result for varied document types as the SMS messages. In fact, other document clustering applications require several preprocessing procedures which provide significant improvements in its performance.

3. Research Method

The design and methods used in the parsing algorithm, language translator, implementation, and performance evaluation in this study are based on the network architectural model shown in Figure 3. The process begins when the system receives the SMS message through a mobile broadband device. The SMS message is automatically downloaded and undergoes the text clustering preprocessing technique.

3.1. The Proposed Technique

The text preprocessing procedure used in this study is primarily based on the study of Lat, et al [6]. Moreover, some major components like tokenization, stop word removal, and stemming are based from the most common preprocessing technique of related studies. This proposed procedure has seven (7) stages with language translator to produce the monolingual English version of the mixed *Bisaya* and English words as illustrated in Figure 4. In [17] this study, the term frequency co-occurrence clustering approach of is implemented and the percent error in the clustered SMS messages is computed to determine and test the accuracy of the preprocessing technique.

3.2. Tokenization, Language Tagging, and Stop-word Removal

Tokenization, language tagging, and stop-word removal are the first three (3) essential steps in the text preprocessing procedure. Tokenization initially breaks the SMS message into tokens based on a set of pre-defined delimiters. To achieve this, an improved greedy algorithm of shown in Figure 1, was enhanced in this study to generate more quality tokens. This method was done by incorporating an additional procedure in the algorithm which either throws away or retains the symbols, digits and punctuation marks in a token, as described in the algorithm shown in Figure 5.

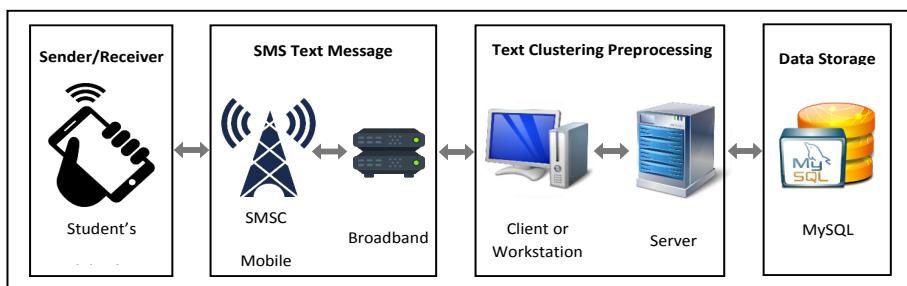


FIGURE 3. System network architectural model.

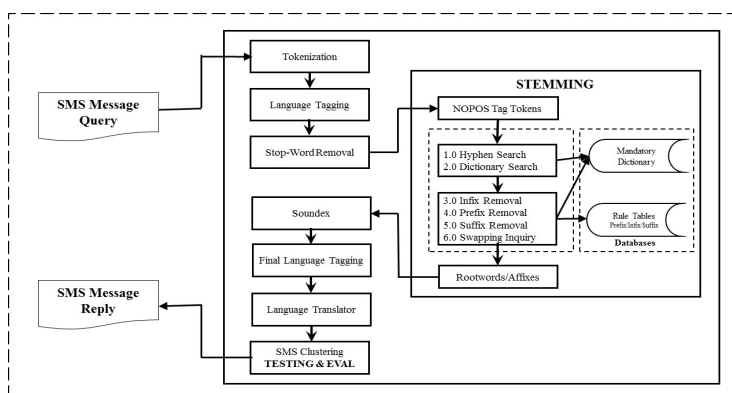


FIGURE 4. The system architecture of text preprocessing technique.

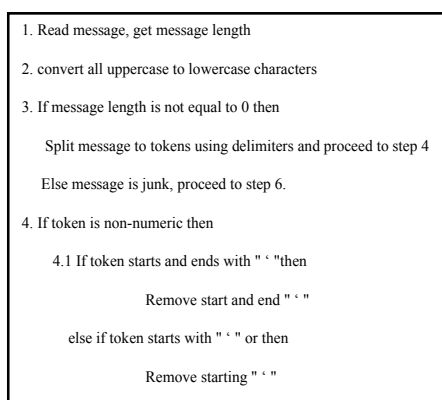


FIGURE 5. Enhancement of an improved greedy algorithm.

A language tagging algorithm based on the *single approach tagging model* introduced by Pushplata [12] is then incorporated for early language detection of tokens to minimize the application of stemming task in the message. This single approach tagging model has only one table tag structure for pair of languages which serves as dictionary lookup for the tagging of tokens. Recurring meaningless words called *stop words*, which were already identified during the language tagging step, are directly removed from the compiled words using the *stop list*, a list of words which serves as the lookup table in removing the stop words in the message. This method differs from the traditional stop word algorithm in which stop words are only removed later in the process.

3.3. Stemming

Stemming, the process of extracting relevant keywords, is another essential procedure done in this study. To achieve this, the Tagalog Stemming Algorithm (TagSA) [2] was enhanced by incorporating the *Bisaya* language morphological features into all stages of its stemming routines as well as an enrollment inquiry setting specifically dealing with tuition fee, down payment, enrollment period, courses offered, and requirement inquiries

for HEIs. Hence, the proposed stemming algorithm is called *BisEng Stemming Algorithm*. Here, the tokens undergo six (6) stemming routines to extract root word or affixes as illustrated in Figure 6.

The first two (2) routines (1.0 and 2.0) removed the hyphenated prefixes from the tokens. The mandatory (root) keywords database serves as the lookup for the closest match of the possible root when the hyphen is removed or retained. The remaining word is considered as the root which will immediately undergo language tagging or may still undergo the next routines. The succeeding three (3) routines (3.0, 4.0, and 5.0) perform infix, prefix, and suffix removal procedures with rule applications stored in the Rule Table Database. The rules in these routines are grouped into sections corresponding to the initial letter of the prefix, succeeding characters in the prefix, and the final letter of the suffix. The last routine, 6.0, is an additional procedure called *swapping inquiry* routine. This step takes place when none of the prefix and suffix removal is applicable but rather a combined prefix and suffix or infix and suffix removal obtains a root word. This procedure will cause the original word to be “recalled” and “reprocessed” alternately with prefix and suffix routines until rules and conditions are satisfied, and root word or affix is produced as an input to the succeeding stages of preprocessing.

It should be noted that *Bisaya* dialects barely have less than three (3) character roots. Thus, some restrictions applied from TagSA are enhanced within the prefix and suffix removal routine, as follows.

- a) If the form starts with a vowel, then at least three letters must remain after stemming, and at least one of these must be a consonant.
- b) If the form starts with a consonant, then at least three characters must remain after stemming, and at least one of these letters must be a vowel.

Phoneme change may either occur in *Bisaya* prefixation or suffixation. In this study, these are the conditions enhanced from TagSA:

For Prefix Removal: If the phoneme *-l/-* or *-d/-* appears in the initial position of the form preceded by an affix that ends in a vowel and is followed by a vowel, then replace it with phoneme *-lh-*. If a consonant *-g-* appears after the first consonant such as *pg-*, *mg-*, *tg-* in the initial position of the word, then insert *-a-* within these two (2) consonants.

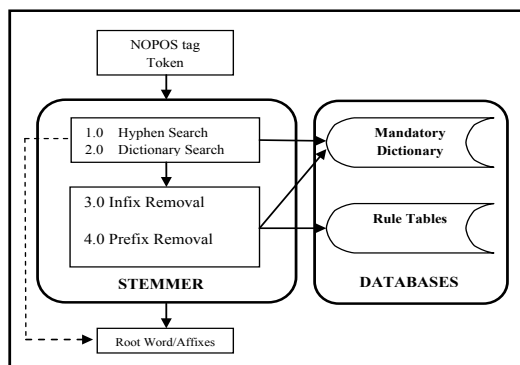


FIGURE 6. Architectural design of BisEng stemming algorithm.

For Suffix Removal: If the phoneme *-u-* appears before the last consonant of the form, then replace it with the phoneme *-o-*. If the phoneme *-ro-* appears before the final consonant of the form, then replace it with the phoneme *-ra-*.

3.4. Soundex Algorithm, Final Tagging and Language Translator

The final three (3) processes in text preprocessing techniques are the implementation of the Soundex algorithm, final tagging, and the use of language translator. The Soundex algorithm is the most widely used technique for correcting misspelled words, character deletion/dropped and abbreviation problems called *Noisy Words*. The final language tagging is used to increase the accuracy of tags since a word may not exist in the dictionary but its root word might [6]. The language translator implemented a baseline approach called *document glossing* [5] to translate all Bisayan tokens into English since all mandatory terms in the clusters are all English. A simple dictionary lookup is done for every *Bisaya* tokens to determine its English translation directly applying the simple procedure is a proof-of-concept that complex translation is unnecessary for clustering.

3.4.1. Application of Term Frequency Co-occurrence in Clustering SMS Messages

The term frequency co-occurrence [17] was used in this study to determine corresponding SMS message clusters. Term matching determines the term frequency. Each query term from an SMS message compared its similarities with the terms in each cluster of mandatory terms such as shown in Table 1. A word has account of “1” if its query term matches or is similar with the text document cluster in Table 1. Otherwise, terms with no matches are marked with “0”. The summary of term matching results is shown in Tables 2 and 3. The representation of the similarities does not consider the ordering of the query term in the cluster since the grammatical structure is not covered in this study.

The similarity score between the query message and the clusters are computed using the formula shown below.

$$\text{Similarity Scores } (q, cg) = \frac{|C(q) \cap C(cg)|}{|C(q)| + |C(cg)| - |C(q) \cap C(cg)|} \quad (1)$$

where $C(q)$ = term frequencies at query terms; $C(cg)$ = term frequencies of cluster group; $C(q) \cap C(cg)$ = term frequencies of matched terms.

The similarity scores range from 0 to 1 and the term frequency cannot be negative. The highest computed similarity scores obtained using (1) corresponds to the cluster document where the query message belongs and sends its assigned message reply.

3.5. Evaluation of the Proposed Technique

This study is implemented in an automated SMS message reply system to measure the performance and quality of the proposed preprocessing technique. The system is using the

TABLE 1. Text document cluster

Cluster	Mandatory terms			SMS message reply
A – Tuition fees	enroll	downpayment	enrollment	Thank you for your inquiry. You can be enrolled at a minimum downpayment of P1,000 only.
	registration	pay	cost	
	payment	how	tuition	
	pass	entrance	passer	
	fee	scholar	unit	
B – Requirements	much	down		Thank you for your inquiry. For freshman, pls bring your high school card, good moral cert., NSO birth cert, and 2x2 ID pic, while TOR, hon. dismissal, good moral cert., NSO birth cert, and 2x2 ID pic for transferees.
	enroll	transferee	enrollment	
	submit	what	location	
	requirement	bring	freshman	
	start	go	class	
C – Enrollment period	where	period	until	Thank you for your inquiry. Enrollment starts 2nd week of October from Monday to Saturday, 8 am–6 pm at COC Carmen/Puerto and will end on Nov 15. Classes start Nov 11.
	late	schedule	school	
	what	end enroll		
	when	enrollment	go	
	deadline	where		
D – Courses offered	course	Electrical	what	Thank you for your inquiry. COC-Phinma Offers Mass Comm, Criminology & Criminal Justice, Education, Engineering, Architecture, Institute Tech. , HRM, Info Tech, Nursing, Mngt & Accounting, Graduate School and Short-term Courses
	Electronics	offer	CivilEng	
	Accountancy	HotelResMngt		
	Architecture	MassCom	BusinessAd	
	Engineering	InfoTech	ComEng	
E – Enrollment exams	Nursing	Mechanical	Sciences	Thank you for your inquiry. No entrance exams required for freshmen and transferees. All incoming freshmen are invited to take the scholarship exams for every Sat from 8 am to 10 am from Oct 19 to Nov 9.
	Computer	ComEng	Education	
		Tourism		
	ElecComEng			
	entrance	grade	schedule	
	scholarship	exam	fee	
	average	scholarship		

TABLE 2. Formal SMS messages test results on term frequency co-occurrence clustering

Simple				Moderate				Difficult			
SMSNO.	Test 1	Test 2	Test 3	SMSNo.	Test 1	Test 2	Test 3	SMSNo.	Test 1	Test 2	Test 3
SMS1	1	1	1	SMS31	1	1	1	SMS61	1	1	1
SMS2	1	1	1	SMS32	1	0	0	SMS62	1	1	1
SMS3	1	1	1	SMS33	1	1	1	SMS63	1	1	1
SMS4	1	1	1	SMS34	1	1	1	SMS64	1	1	1
SMS5	1	1	1	SMS35	1	1	1	SMS65	1	1	1
SMS6	1	1	1	SMS36	1	1	1	SMS66	1	1	1
SMS7	1	1	1	SMS37	1	1	1	SMS67	1	1	1
SMS8	1	1	1	SMS38	1	1	1	SMS68	1	0	0
SMS9	1	1	1	SMS39	1	1	1	SMS69	0	1	1
SMS10	1	1	1	SMS40	1	1	1	SMS70	1	1	1
SMS11	1	1	1	SMS41	1	1	1	SMS71	1	1	1
SMS12	1	1	1	SMS42	1	1	1	SMS72	1	1	1
SMS13	1	1	1	SMS43	1	0	0	SMS73	1	1	1
SMS14	1	1	1	SMS44	1	1	1	SMS74	1	1	1
SMS15	1	1	1	SMS45	1	1	1	SMS75	1	1	1
SMS16	1	1	1	SMS46	1	1	1	SMS76	1	1	1
SMS17	1	1	1	SMS47	1	1	1	SMS77	0	0	0
SMS18	1	1	1	SMS48	1	1	1	SMS78	1	1	1
SMS19	1	1	1	SMS49	1	1	1	SMS79	1	1	1
SMS20	1	1	1	SMS50	1	1	1	SMS80	1	1	1
SMS21	1	1	1	SMS51	1	1	1	SMS81	1	1	1
SMS22	1	1	1	SMS52	1	1	1	SMS82	1	1	1
SMS23	1	1	1	SMS53	1	1	1	SMS83	1	1	1
SMS24	1	1	1	SMS54	1	1	1	SMS84	1	1	1
SMS25	1	1	1	SMS55	1	1	1	SMS85	1	1	1
SMS26	1	1	1	SMS56	1	1	1	SMS86	1	1	1
SMS27	1	1	1	SMS57	1	1	1	SMS87	1	1	1
SMS28	1	1	1	SMS58	1	1	1	SMS88	0	0	0
SMS29	1	1	1	SMS59	1	1	1	SMS89	0	0	0
SMS30	1	1	1	SMS60	1	1	1	SMS90	1	1	1

Legend: 0 – incorrectly clustered; 1 – correctly clustered.

MS Visual Studio (VS) 2008 that runs in Windows XP Service Pack 2, Windows.Net 3.5 Service Pack 3.5 up to Windows 7 platform and database used is SQL Server 2005 express edition or 2008 R2 express edition for workstations or stand-alone implementation. Moreover, Windows Server 2008 with SQL Server 2005 or 2008 R2 Standard database was used as a server in a network-based application.

Actual SMS message exchange between the automated SMS message reply system and the SMS queries allowed the system architecture to significantly evaluate the precision and effectiveness of the algorithms implemented in the preprocessing technique. This study was tested with 180 SMS messages in three (3) trials with three (3) different levels of complexity such as:

- Simple. These are SMS messages with 1–4 terms with one (1) topic inquiry
- Moderate. These are SMS messages with 5–8 terms with at most two (2) topic inquiries
- Difficult. These are SMS messages with more than 8 terms with at most three (3) topic inquiries

TABLE 3. Informal SMS messages test results on term frequency co-occurrence clustering

Simple				Moderate				Difficult			
SMSNo.	Test 1	Test 2	Test 3	SMSNo.	Test 1	Test 2	Test 3	SMSNo.	Test 1	Test 2	Test 3
SMS1	1	1	1	SMS31	1	1	1	SMS61	1	1	1
SMS2	1	1	1	SMS32	1	1	1	SMS62	1	1	1
SMS3	0	1	1	SMS33	1	1	1	SMS63	0	1	1
SMS4	1	1	1	SMS34	1	1	1	SMS64	1	1	1
SMS5	1	1	1	SMS35	1	1	1	SMS65	1	1	1
SMS6	1	1	1	SMS36	1	1	1	SMS66	0	1	1
SMS7	1	1	1	SMS37	0	0	0	SMS67	1	1	1
SMS8	1	1	1	SMS38	1	1	1	SMS68	0	1	1
SMS9	1	1	1	SMS39	1	1	1	SMS69	1	1	1
SMS10	1	1	1	SMS40	1	1	1	SMS70	1	1	1
SMS11	1	1	1	SMS41	0	0	0	SMS71	1	1	1
SMS12	1	1	1	SMS42	1	1	1	SMS72	1	1	1
SMS13	1	1	1	SMS43	1	1	1	SMS73	0	0	0
SMS14	1	1	1	SMS44	1	1	1	SMS74	1	1	1
SMS15	1	1	1	SMS45	1	1	1	SMS75	1	1	1
SMS16	1	1	1	SMS46	1	1	1	SMS76	1	1	1
SMS17	1	1	1	SMS47	1	1	1	SMS77	1	1	1
SMS18	1	1	1	SMS48	1	1	1	SMS78	1	1	1
SMS19	1	1	1	SMS49	0	1	1	SMS79	1	1	1
SMS20	1	1	1	SMS50	1	1	1	SMS80	1	1	1
SMS21	0	1	1	SMS51	1	1	1	SMS81	1	1	1
SMS22	1	1	1	SMS52	1	1	1	SMS82	1	1	1
SMS23	1	1	1	SMS53	1	1	1	SMS83	1	1	1
SMS24	1	1	1	SMS54	1	1	1	SMS84	1	1	1
SMS25	1	1	1	SMS55	1	1	1	SMS85	1	1	1
SMS26	1	1	1	SMS56	1	1	1	SMS86	1	1	1
SMS27	1	1	1	SMS57	1	1	1	SMS87	1	1	1
SMS28	1	1	1	SMS58	1	1	1	SMS88	0	1	1
SMS29	1	1	1	SMS59	1	1	1	SMS89	0	0	0
SMS30	1	1	1	SMS60	1	1	1	SMS90	1	1	1

Legend: 0 – incorrectly clustered; 1 – correctly clustered.

Each level of complexity is computed with percent error with the following test classifications:

- Formal SMS. These are SMS messages with standard and complete text typing styles
- Informal SMS. These are SMS messages with varied text typing styles.

Frequently asked questions on enrollment inquiries which serve as test data are provided by one of the HEIs in the city, Cagayan de Oro College-Phinma, from the year 2013 and 2014's manual SMS enrollment campaign data.

The formulas used for the computation of the percent errors are defined as:

Percent Error on Text Preprocessing (*PETP*):

$$PETP = \left[\frac{ErPrepQ}{TMsgQ} \right] \times 100 \quad (2)$$

where $ErPrepQ$ = the total number of erroneously preprocessed SMS queries; $TMsgQ$ = the total number of SMS message queries

Percent Error on Clustering (PEC):

$$PEC = \left[\frac{ErCMsg}{TPrepMsg} \right] \times 100 \quad (3)$$

where $ErCMsg$ = the total number of erroneously clustered SMS messages; $TPrepMsg$ = the total number of preprocessed SMS message

Average Percent Error on Text Preprocessing ($AvePETPrep$):

$$AvePETPrep = \frac{PETP_1 + PETP_2 + PETP_3}{3} \quad (4)$$

where $PETP_i$ = Percent Error on Text Preprocessing per Category; $i = 1, 2, 3$.

Average Percent Error on Clustering ($AvePEC$):

$$AvePEC = \frac{PEC_1 + PEC_2 + PEC_3}{3} \quad (5)$$

where PEC_i = Percent Error on Clustering per Category; $i = 1, 2, 3$.

The computed percent error values are between 0 and 100; a lower value indicates a higher accuracy rate on significant term extraction. The average processing time for each message as shown in Table 5, are also evaluated to determine the lead time between the text preprocessing and clustering process of the system.

4. Results and Discussions

The preprocessing technique was evaluated by percent error computation on the SMS message term extraction through term frequency co-occurrence clustering. The term frequency co-occurrence clustering approach further evaluates if the term extraction using the text preprocessing technique is effective to identify the SMS message cluster(s) as presented in Tables 2 and 3 below. Each SMS message query term is evaluated by comparing its similarities with the terms in each cluster of the mandatory terms such as shown in Table 1. The highest computed similarity scores obtained using (1) corresponds to the cluster document where the query message belongs and sends its assigned message reply. Each SMS message correctly clustered with correct SMS message reply is marked "1", otherwise, it is marked with "0", as shown in Tables 2 and 3.

Table 4 presents the summary of the text preprocessing error rate result after consolidating detailed formal and informal SMS messages test results on text preprocessing. The errors occur on informal messages are due to the following factors:

TABLE 4. Summary of percent error result on SMS message text preprocessing (PETP)

Complexity level	Test 1		Test 2		Test 3		Average percent error per category (AvePETPrep)
	Raw score	PETP ₁	Raw score	PETP ₂	Raw score	PETP ₃	
Formal messages							
Simple	1/30	3%	1/30	3%	1/30	3%	3%
Moderate	3/30	10%	6/30	20%	6/30	20%	17%
Difficult	6/30	20%	7/30	23%	7/30	23%	22%
Informal messages							
Simple	4/30	13%	2/30	7%	2/30	7%	9%
Moderate	10/30	33%	7/30	23%	7/30	23%	27%
Difficult	10/30	33%	4/30	13%	4/30	13%	20%

4.1. Spelling Correction (Soundex) Errors

These errors are primarily due to the presence of abbreviations and replaced characters with digits in the test data, which are among the constraints in this study. These abbreviated and varied text style keywords are too short to handle by the Soundex or any other algorithm to provide the complete keywords. These keywords include the following:

reqt for requirement

l8t for late

dwn for downpayment

educ for education

ddlyn for deadline

ave for average

wer for where

Though “*reqt*” is an abbreviation and a limitation of the study, it is still considered a significant keyword because it is frequently used on inquiries rather than the lengthy word “*requirement*”. This keyword will somehow create an effect on the accuracy of clustering informal messages if this keyword is disregarded. After careful study, this keyword is then added to the dictionary and associated with its complete word “*requirement*”.

As the keyword “*reqt*” is added in the second testing, it reflects an error rate drop off of 10% on informal-moderate and 20% on informal-difficult messages. Even though there is only a slight 10% error rate increase on the formal-moderate and 3% on formal-difficult messages, there is no significant effect on the preprocessing stage in the actual marketing campaign for the school. Since almost all of the SMS messages received uses the keyword “*reqt*” rather than the lengthy “*requirement*” keyword.

4.2. Language Translation Errors

It is observed that if keywords are erroneously stemmed and/or Soundex, error cascades to language translation. For instance, the keyword “*madayon*” is stemmed to “*dayon*”, and its closest keyword match in the dictionary for Soundex is “*dalhon*”. In turn, the keyword “*dalhon*” is translated to “bring” instead of “continue” for “*dayon*”.

4.3. Limitation on Grammatical Structures

Since this study does not cover grammatical structure, the preprocessing technique does not capture sentence patterns. Therefore, the system restricts double entry of the same keyword with different meanings concerning its part of speech (noun, pronoun, verb, adjective, adverbs, etc.).

Table 4 also shows that formal-moderate and formal-difficult messages have an average error rate of only 17%–22%, while 20%–27% on informal-moderate and informal-difficult messages. Therefore, approximately between 17% and 27% overall error rate on text preprocessing technique. It is expected that informal messages have higher error rate compared to formal messages since it is more complex in nature and errors are primarily caused by the limitations as mentioned above on grammars, abbreviations, digits, and days.

Table 5 presents a summary of the SMS message term frequency co-occurrence approach clustering error rate after consolidating detailed formal and informal SMS messages test results on term frequency co-occurrence approach clustering from Tables 2 and 3. It is clearly shown that the error rate of clustering is dependent on the error rate of the text preprocessing. As the text preprocessing error rate decreases, clustering error rate also decreases. This only proves that wrong extraction of keywords has a significant impact on the efficiency of clustering. Moreover, it is also evident in the average percent error rate that the text processing technique is effective with approximately 10%–13% average error rate between formal and informal messages when text preprocessing is applied to clustering.

Table 6 shows the average processing time performance of the test data. It is clearly demonstrated that there is only a minimal time difference in processing various message types. The difference varies on the number of text preprocessing procedure applied as the level of message difficulty increases. Furthermore, simultaneous texting is also done during testing and it does not show any effect on the performance of the preprocessing technique and processing time of the system since arrival time of messages in the system varies on the response time of the mobile network provider of the sender.

The above results indicate that the proposed text preprocessing technique effectively addressed keyword extraction and text clustering challenges for multilingual documents

TABLE 5. Summary of percent error result on SMS message clustering

Complexity level	Test 1		Test 2		Test 3		Average percent error per category (AvePEC)
	Raw score	PEC ₁	Raw score	PEC ₂	Raw score	PEC ₃	
<i>Formal messages</i>							
Simple	0/30	0%	0/30	0%	0/30	0%	0%
Moderate	0/31	0%	2/31	7%	2/31	7%	4%
Difficult	4/30	13%	4/30	13%	4/30	13%	13%
<i>Informal messages</i>							
Simple	2/30	7%	0/30	0%	0/30	0%	2%
Moderate	3/30	17%	2/30	7%	2/30	7%	10%
Difficult	6/30	20%	2/30	7%	2/30	7%	11%

TABLE 6. Text clustering preprocessing average processing time performance

Complexity level	Average processing time performance in milliseconds		
	Test 1	Test 2	Test 3
<i>Formal messages</i>			
Simple	11	14	18
Moderate	14	16	16
Difficult	17	19	20
<i>Informal messages</i>			
Simple	11	13	16
Moderate	15	19	19
Difficult	19	19	19

on SMS messages, specifically for mixed *Bisaya* and English terms. It also generates an effective text preprocessing technique which can be used as basis for an automated SMS response for HEI enrollment related inquiries. Using the method proposed in this study, any academic institution may be able to maximize the opportunity to effectively entertain more SMS enrollment related inquiries as alternative communication medium to its target market. Furthermore, this showcases technological advancement of the institution as it utilizes an ICT-enhanced marketing approach through mobile technology. The parents, guardians, and students, on the other hand, will have a first-hand access on enrollment information via mobile phones when internet is unavailable to access the website or phone call loads are insufficient.

5. Conclusion, Limitation, and Further Research

In this study, a text clustering preprocessing technique for mixed *Bisaya* and English SMS message for HEI enrollment-related inquiries is successfully developed. The technique is relatively a new approach in extracting significant keywords while addressing key challenges on mixed *Bisaya* and English SMS messages. It also demonstrates a good preprocessing procedure for morphologically complex *Bisaya* language structure with combined affixes.

Test results revealed that adopting the technique can effectively extract essential keywords with approximately 17%–27% overall average error rates on text preprocessing and 10%–13% error rates when applied to baseline approach clustering. This means a roughly 73%–83% accuracy rate on text processing and an 87%–90% accuracy rate is achieved when text preprocessing is applied to clustering. The errors are primarily caused by abbreviations, days, dates, numerical value replacement of characters in a keyword and grammar structures which are limitations of this study.

Based on the findings of this study, it is safe to conclude that the designed algorithms generate an effective text preprocessing technique for mixed *Bisaya* and English SMS messages which can be used as the basis for an automated response for HEI enrollment-related inquiries. Using the method proposed in this study, any academic institution may be able to maximize the opportunity to effectively entertain more SMS enrollment-related

inquiries as alternative communication medium to its target market. Furthermore, this showcases technological advancement of the institution as it utilizes an ICT-enhanced marketing approach through mobile technology.

In future studies, it is suggested to improve the text preprocessing algorithm which can extract and identify abbreviations, days, dates, tuition fee inquiries and numerical value character replacements; and implement a more intensive clustering algorithm which caters grammar to strengthen the accuracy of automated replies on the SMS message enrollment inquiries.

References

1. Yogatama D. Clustering multilingual documents by estimating text-to-text semantic relatedness. 2010; 1–29.
2. Katariya MNP, Chaudhari MS, Subhani B, Laxminarayana G, Matey K, Nikose MA, Deshpande SP. Text preprocessing for text mining using side information. *International Journal of Computer Science and Mobile Applications*. 2015; 3(1), 01–05.
3. Leong CK, Lee YH, Mak WK. Mining sentiments in SMS texts for teaching evaluation. *Expert Systems with Applications: An International Journal*. 2012; 39(3), 2584–2589.
4. Froud H, Lachkar A, Ouatic SA. Arabic text summarization based on latent semantic analysis to enhance Arabic documents clustering. *International Journal of Data Mining & Knowledge Management Process*. 2013; 3(1), 79–95.
5. Evans DK, Klavans J. *A platform for multilingual news summarization*. Computer science technical report. University of Columbia: New York. 2003; 3–4.
6. Lat JO, SzeK, Ng ST, Yu GD, Lim NRT. Lexicon acquisition for the English and Filipino language. Proceeding of the 3rd national natural language processing research symposium, Manila, Philippines. 2007, 2–3.
7. Blake F. Differences between Tagalog and Bisayan. *Journal of the American Oriental Society*. 1994; 25, 162–169.
8. Bonus DEJ. *The Tagalog Stemming Algorithm (TagSA)*. Proceeding of the national natural language processing research symposium, Manila, Philippines. 2004, 63–67.
9. Ramasubramanian C, Ramya R. Effective pre-processing activities in text mining using improved porter's stemming algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*. 2013; 2(12), 2278–1021.
10. Morik K, Martin S. The Mining Mart approach to knowledge discovery in databases. *Intelligent Technologies for Information Analysis: Springer Berlin Heidelberg*. 2004, 47–65.
11. Naseem T, Snyder B, Eisenstein J, Barzilay R. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*. 2009; 36(1), 341–385.
12. Pushplata RC. *An analytical assessment on document clustering*. *International Journal of Computer Network and Information Security*. 2012; 4(5), 63–71.
13. Schmid H. Tokenizing – computational linguistics and phonetics. In: *Corpus linguistics: an international handbook*. Walter de Gruyter: Berlin. 2007; 2–4.
14. Han J. Building an efficient, scalable and trainable probability-and-rule based part of speech tagger of high accuracy. (Doctoral dissertation, uga). 2009; 6–39.
15. Khan O, Karim A. MIKE: An interactive micro blogging keyword extractor using contextual semantic smoothing. Proceedings of 24th international conference on computational linguistics: COLING (Demos). 2012, 289–296.

16. Nayak AS, Kanive AP, Chandavekar B, Balasubramani R. Survey on pre-processing techniques for text mining. *International Journal of Engineering and Computer Science*. 2016; 5(6), 16875–16879.
17. Kaji H, Aizono T. Extracting word correspondences from bilingual corpora based on word co-occurrences information. *Proceeding of the 16th conference on computational linguistics-volume 1*. Association for Computational Linguistics. 1996, 1, 26–28.