NORM: A Text Normalization System for Filipino Shortcut Texts Using the Dictionary Substitution Approach

Gems Cuevas, Jedd Gopez, Nicco Nocon, Peter Suministrado
College of Computer Studies
De La Salle University - Manila
{justingemscuevas, gopez.jedd, noconoccin, peter.suministrado}@gmail.com

ABSTRACT
The popularity of using mobile devices for communication leads to trends on modification of language particularly shortcut texting, which extends to the Internet world. In Natural Language Processing (NLP), data gathered comes from various sources such as social media, which may consist of shortcuts that hinder existing tools from working properly. In this paper, we present NORM, a system that normalizes shortcut texts in the Filipino language using the dictionary substitution approach. Moreover, randomization and word frequency count was considered in case there are multiple matches in the dictionary. The study involved a total of 4,000 word entries: 2,250 for the word dictionary, 500 for testing, and 1,250 for the corpus used to determine the frequency count.

General Terms
Performance, Experimentation, Languages

Keywords
Normalization, Filipino, Shortcut texts, Preprocessing

1. INTRODUCTION
According to Entrepreneur Philippines[1], the Philippines is a growing technologically capable and up-to-date users with over 33M Internet users and 106M mobile users, as of the year 2012. The modification of words using symbols and other combination of characters to minimize the word length has been a trend that was started by people that uses the Internet and mobile devices; thus, altering and deterring the word's formation as well as meaning [1]. Studies about the origins on how to create shortcut texts were done by many researchers. However, most of these studies only cover the English language. Since the usage of Filipino is almost nationwide and that is commonly spoken in urban cities, this research presents NORM, a system that normalizes shortcut texts having the Filipino language as its domain.

For the English language, different authors [8, 9] discovered different categories of styles, such as: shortenings, omitting middle letters, acronyms and initialisms, number and letter homophones, and more. For the Filipino language [3], there are some that are similar with the styles in the English language, for instance is the phonetic style, but due to the structures and complexity of the language, additional styles emerged like ending 'a', consonant skeleton and repeating of units. Furthermore, the language created phenomena such as: code-switching, where two languages intertwine but create context; aspect, in the context of the Filipino language shows similar treatment of verbs when providing more structure to tenses; and affixation reduplication, words morphed by adding affixes, providing more detail (e.g. "kuha", "kinuha", "kukuha", "kinukuha", "kumukuha").

Text normalization refers to the process of transforming the shortcut text back into its standard form. Existing normalization systems particularly the works of [2, 5, 7] are currently available only for the English language. This shows an excellent opportunity to have a comparable translator for the Filipino Language for the reason that Filipinos – as being part of social networks, forums, chatting applications, and SMS messaging systems – tend to shorten messages and mostly words for faster and memory-efficient responses.

Normalizing a text will also be useful to Natural Language Processing (NLP) because text collected for data mining might not be in their normal form – data gathered comes from different Internet sources which contains shortcut texts – thus, hindering existing data processing tools to fully and completely process the data. Moreover, instead of having texts that are not in their normal form, NORM can act as a form of cleaning in removing such types.

2. RELATED WORKS
2.1 Dictionary Substitution Approach
The Dictionary Substitution (DSA) or Search & Replace approach by Raghunathan and Krawczyk [7] is commonly used by web-based systems that normalize shortcut texts such as Transl8it[2] and Lingo2word[3]. The process of the approach is to simply find the word to be substituted in the dictionary. The word is substituted upon a match. Otherwise, it is retained. When there are multiple matches, the selection process in substituting is randomized.

Raghunathan and Krawczyk built three dictionaries for testing: a native dictionary containing 560 unique Short Messaging Service (SMS) token entries made up of abbreviated or SMS message colloquialisms from their training data; a "web-small" having 224 SMS entries from Transl8it and Lingo2word; and a "web-big" comprising 1,611 entries from a fusion of the previous two internet sources and Dtxtrapp[4]. Additionally, an empirical substituted set was also made, which is based on the repetitiveness of substitution from the training data.

The evaluation results of their study are shown in Table 1, where they tested their system through three test sets from National University of Singapore (NUS), Hong Kong University (HKU),

[1] From http://www.entrepreneur.com.ph/ideas-and-opportunities/article/good-to-know-philippine-web-statistics/

[2] From http://transl8it.com/

[3] From http://www.lingo2word.com/

[4] From http://www.dtxtrapp.com/

Proceedings of the 10th National Natural Language Processing Research Symposium, pages 86–91
De La Salle University, Manila, February 21-22, 2014.

and TreasureMyText (TMT). They tested their system using the four dictionaries which is nicknamed in the table namely native-random, web-small, web-big, and native-empirical, respectively. The results show that native dictionary performed better than the internet dictionaries and the empirical native dictionary was the best among all of the dictionaries, indicating that common shortcuts have a higher potential in getting used than the others.

Table 1. Performance of Raghunathan and Krawczyk's Dictionary Substitution Approach

System	NUS Test Set	HKU Test Set	TMT Test Set
	BLEU	BLEU	BLEU
Baseline	0.562	0.7025	0.4009
Web-Small	0.6488	0.7218	0.4933
Web-Big	0.5573	0.8128	0.4311
Native-Emp	0.8941	0.8770	0.5873
Native-Random	0.7945	0.7940	0.5335

In using DSA, there are some problems that arise such as ambiguity and fixed translations. In ambiguity, shortcuts with multiple meanings might result into an incorrect translation. An example is the shortcut *2* which has three counterparts namely *two*, *to* and *too*. The dictionary substitution approach would not distinguish which of the three would be the correct translation, resulting in an arbitrary translation. Another problem is when a shortcut contains a fixed translation in the dictionary. Using the same example for the previous problem, the shortcut *2* has multiple counterparts; if the dictionary indicates *two* as its only counterpart, then *two* will always be the result for the translation – creating incorrect sentences like "I will go *two* the mall." Additionally, problems on the lack of entries in dictionaries and misinterpretation of the user input are included.

2.2 Generic Letter Transformation

A Generic Letter Transformation approach by [5] is developed to transform shortcut texts to its normalized forms without explicitly categorizing the non-standard tokens into insertion, substitution, or deletion. It is a rule based approach that does not return back the normalized word immediately rather transforms a non-standard token to a standardized form of that token which at the end, can be concatenated to form a normalized word. A non-standard token can be labelled with the following to form a normalized English word (a) one of the 0–9 digits, (b) one of the 26 alphabets including itself, (c) the null character represented by "-", and lastly (d) a letter combination; a sequence labelling framework automatically carries out this process. Acronyms are one of the exceptions in this experiment.

For the data collection, they utilized the Edinburgh Twitter corpus which contains 97 million Twitter messages. They extracted the English tweets only through the use of TextCat, a language identification toolkit. A total of 62,907 training word pairs were generated after this process. They made three variations of their experiment (a) LetterTran (All) which generates up to 30 variants for a given non-standard word, (b) LetterTran (Trim) where the lookup table is trimmed by choosing only the most frequent dictionary words and their generated variants, and (c) LetterTran (All) + Jazzy where the Jazzy spell checker is integrated with the LetterTran (All) allowing the system to return an output whenever the LetterTran (All)'s candidates are not confident.

In evaluating the system, 303 pairs of SMS messages and 3,802 pairs of Twitter messages were manually gathered by the researchers. It is then compared to three approaches: (a) a list of chat slangs and acronyms collected by InternetSlang[5], (b) a word-abbreviated lookup table generated by the supervised deletion-based abbreviation approach by Pennel and Liu in 2010, and (c) the jazzy spell checker by Idzelis in 2005. The system is then measured through the use of the n-best accuracy where the n is one and three. The results in Table 2 show how the LetterTran systems outperform the other systems compared to it. It also shows how it performed better given a set of data from Twitter since the data for training came from Twitter itself.

Table 2. N-best performance of Generic Letter Transformation

System Accuracy	Twitter (3802 pairs)		SMS (303 pairs)	
	1-best	3-best	1-best	3-best
InternetSlang	7.94	8.07	4.95	4.95
Pennel et al. 2010	20.02	27.09	21.12	28.05
Jazzy Spell Checker	47.19	56.92	43.89	55.45
LetterTran (Trim)	**57.44**	**64.89**	**58.09**	**70.63**
LetterTran (All)	**59.15**	**67.02**	**58.09**	**70.96**
LetterTran (All) + Jazzy	**68.88**	**78.27**	**62.05**	**75.91**
Choudhury et al. 2007	n/a	n/a	59.9	n/a
Cook et al. 2009	n/a	n/a	59.4	n/a

2.3 Noisy Channel Model

The noisy channel model by Brill and Moore [2] has been used in different types of problems, including spelling corrections. The model has two components: the source model and the channel model (also referred to as the error model). The problem that they were trying to address is on automatically training a system to correct generic single word spelling errors, with exception to the possibilities of word confusions (e.g. two, too and to).

The model works by learning generic string to string edits, along with the probabilities (Bayes' rule) of each of these edits – to further improve its accuracy. The approach created returns an n-best list of candidates according to the error model, and then rescored by taking into account the source probabilities.

The experiment was done using a 10,000 word corpus of common English spelling errors that are paired with their correct spelling (80:20 rule). The dictionary used contained approximately 200,000 entries which included all words in the test set. Brill and Moore also tried adding a language model that was derived from a large collection of online text. The results were that without a language model, the error model showed a 52% reduction in spelling correction error rate compared to Church and Gale's weighted Damerau-Levenshtein distance technique. With a language model though, an increase showed in spelling correction error rate resulting to 74% reduction.

[5] http://www.internetslang.com/

3. FILIPINO SHORTCUT TEXTING STYLES

In transforming Filipino normalized texts into their shortcut forms, different shortcut styles are followed (see Table 3 for examples). In the study by [3], they found four (4) most-used shortcut styles in the Filipino language – ending 'a', consonant skeleton, phonetic style, and repeating units.

In the Filipino language, one-syllable words are normally used. In order to shorten the use of these one-syllable words even more, the letter 'a' is removed resulting into a one-lettered consonant and the appearance of ending 'a' style. Another style in Filipino is the consonant skeleton where the vowels between consonants are removed. The observation regarding this style is that the beginning and ending letters of the word hold more importance than the other letters, resulting into the removal of vowels in the middle part of the word. For phonetic style, it is basically the replacement of characters that matches the same pronunciation as the original ones – it also includes words from the English language. Another usage of numbers is found on another style in Filipino called repeating units, where syllables of the word are repeated. The structure of this style is the following: [syllable][number of repetition] – when followed transforms *pupunta* (going) into [*pu*][*2*]*nta* or *pu2nta*. Cabatbat and Tapang also included few of the less usual styles that were also found which are the omission of the letter 'h', or omission of 'i' in the beginning of a word.

In the Filipino language, code-switching is included – meaning other languages such as English may be mixed with Tagalog. In the works of [8, 9], they sought and found out different shortcut styles for the English language. Generalizing their findings, the list of shortcut styles are the following: shortenings, contractions, clippings or truncations, initialism or alphabetism, acronyms, letter/number homophones or orthographic transformation, intentional misspellings or non-conventional spellings, accent stylisation, and emoticons.

Shortenings is a type of transformation wherein the latter part of the original word is pruned. In contractions, the main goal is to compress the words by removing vowels or consonants. In clippings or truncations, a word is shortened by removing a letter at either the beginning or end. Another style for English is initialism or alphabetism, which are used to convert the first letters of the related words in phrases, syllables or parts of a word, or combinations of words and syllables to form shortened version and the product is pronounced by the letters instead of the word. Acronyms, on the other hand, are different from initialism for they are created by taking the initial letters of words in order to produce a new pronounceable word. The letter/number homophones or orthographic transformation is another style wherein phonetically similar numbers or letters are used in order to substitute part/s of the original texts. For intentional misspellings or non-conventional spellings, words are shortened by removing or completely changing letters. However, pronunciations stay intact or become a bit dented. An example for misspelling is *excelent* from *excellent*; as for non-conventional spellings, *love* becomes *luv*. Accent stylization is when the word is transformed or styled with the blend of the user's accent like when *give me* becomes *gimme*. Last is the use of emoticons which are series of special characters that depict a face when combined. These are used to convey expressions or emotions – instead of typing "I am *happy*!", the user may express it with the smiley ':-)'.

Table 3. Shortcut Texting Styles and its Examples

FILIPINO		
Style	Normalized Form/s	Shortcut Form
Ending 'a'	na (that)	n
	sa (to)	s
	ba (do)	b
Consonant Skeleton	naman (so)	nmn
	bakit (why)	bkt
	dapat (should)	dpt
Phonetic Style	ito (this)	i2
	siya (he/she)	xa, xya, cya
	dito (here)	d2
Repeating Units	bababa (less/going down)	ba3
	pupunta (going)	pu2nta
	pinagsamasama (gather)	pinagsama2
Omission of 'i' or 'h'	iyong (your)	yong
	kahihintay (waiting)	kaiintay
	ilan (how many)	lan
ENGLISH		
Shortenings	morning	morn
	after	aft
Contractions	week	wk
	next	nxt
Clippings or Truncations	going	goin
	till	til
Initialism or alphabetism	World Health Organization	WHO
	As soon as possible	ASAP
Acronyms	Acute Immune Deficiency Syndrome	AIDS
Letter/number homophones or Orthographic transformation	before	b4
	you	U
Intentional misspellings or non-conventional spellings	night	nite
	love	luv
Accent stylisation	what's up	wassup, wazzup
	cause	cos, coz, cuz
Emoticons or Smileys (not included in the system)	:-)	smiling
	:-(	sad

4. NORMALIZATION OF FILIPINO SHORTCUT TEXTS (NORM)

NORM is a normalization system in a form of a mobile-web application. It is composed of three (3) modules namely the user interface, tokenize module, and normalize module, with word dictionary as its main resource (see Figure 1).

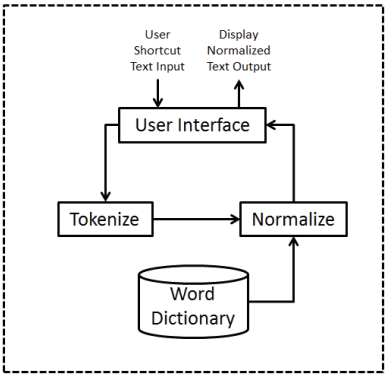

Figure 1. System Architecture

The total number of data built and used for the system is 4,000 Filipino word entries. Table 4 shows the breakdown of training and testing data following the 80:20 rule.

Table 4. Breakdown of Data

Data	Word Entries
Training (80)	2,250
Testing (20)	500
Corpus used to determine frequency count	1,250
TOTAL	4,000

4.1 Word Dictionary

The main resource for the normalization system is the word dictionary. It has a total of 2,250 word entries and is represented in an XML file. An example is shown in Figure 2 where *lang* is translated as *only* in English.

```
<Combinedictionary>
    <entry>
        <input>ln</input>
        <output>lang</output>
        <frequency>20</frequency>
    </entry>
</Combinedictionary>
```

Figure 2. Word Dictionary

4.1.1 XML Representation

The XML file contains three main elements namely, the input, output, and frequency tag. The input tag denotes the shortcut word of an entry while the output tag refers to its normalized form. Lastly, the frequency tag indicates the number of times the normalized word appeared in a separate corpus. This corpus was created for the sole purpose of counting the regularity of each entry. It is a small corpus, containing 1,250 words formed by sentences gathered from Internet sources such as social media, blogs, forums, and online chat rooms.

4.1.2 Dictionary Population

The dictionary was built by: (1) taking examples from social networking sites; (2) using existing entries from Transl8it and Lingo2word; (3) manually converting words from an online repository of Tagalog literary and religious texts [4] to shortcut words; and (4) manually adding entries. The entries gathered for the dictionary contains missing counterparts, either the shortcut word or its normalized counterpart; therefore we provided these to complete an entry. In addition, the dictionary was built by populating it with each variant of the gathered word to be able to cover a vast range of words as much as possible; per normalized word, its shortcut variants is created by following the different styles mentioned by the cited sources. On the other hand, normalized variants per shortcut word are created through their different forms such as past, present and future tenses.

4.2 User Interface

The user interface is the front-end part of the system. It functions as an interaction tool between the user and the back-end of the system. The Normalize Filipino Shortcut Texts (see Figure 3) is the page wherein the system translates the user's given shortcut texts into their corresponding normalized forms. It is connected to the back-end of the system where it runs the different system modules to acquire and display the proper normalized texts.

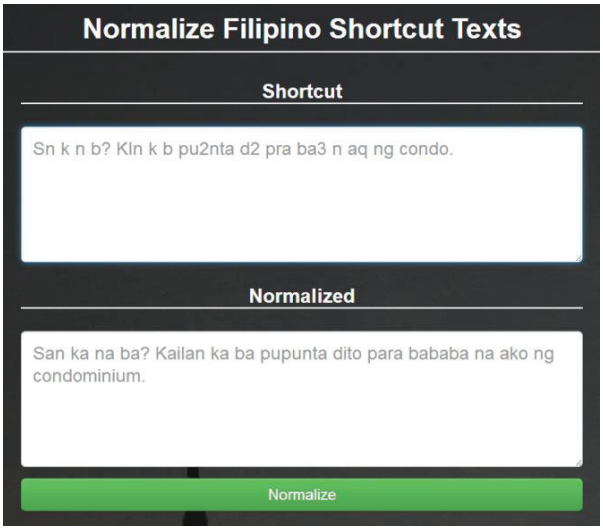

Figure 3. Graphical User Interface

4.3 Tokenize Module

In this module, the input text of the user is tokenized or broken down into words to form "tokens". Before tokenizing, the user's input text is cleaned, meaning the special characters or punctuations are removed from the words. The only punctuations not removed in the input are the hyphen (-) and apostrophe ('), for they are distinguished as part of the word. In tokenizing, spaces are used as delimiter.

4.4 Normalize Module

This module is where the Filipino shortcut text normalization process comes in. It uses the translation rules in the word dictionary using the DSA. When the system does not find a match in the dictionary, the word is retained. In cases wherein there are multiple matches, the system will then determine which to substitute in two ways depending on the chosen type of DSA. One way is to randomize the normalized word to be substituted among the list of matches, and the other is to substitute the entry among the multiple matches with the highest frequency count.

5. TESTING AND DISCUSSION

In testing the system, the translations were done per word and without context (direct translation), for the reason that, the translations are in the same domain or language – making reordering of words in a sentence unnecessary. The system was evaluated using the Bilingual Evaluation Understudy (BLEU) [6], a metric determining the quality of the translated text. These are the results: 0.5780 using randomized DSA and 0.6077 using DSA with frequency. Comparing the two methods, the frequency of words in DSA generated a higher potential in choosing which among the multiple matches are correct. For the reason that, having the high amount of frequency means it is usually or commonly used by people – creating more possibility of it being used in the testing set. It also implies that, even though the system does not give equal chances to the matches on which to substitute, using frequencies made the system smarter by having a heuristic or a form of a basis in choosing which match to substitute.

During the testing stage, several problems were encountered. One of the main problems was the occurrence of multiple matching inputs. Multiple matching inputs lead the system to choose an output among the matches randomly or through the use of frequencies. Filipino shortcut texts contain several shortcut words that when normalized, can have many different translations. Due to the system being unable to acknowledge the context of adjacent words in a given input, there is no other way to know which translation should be used. This, in turn, prompts the randomization method to rely on luck or odds to return the expected output, while the method of using the frequency counts return a fixed output (the one with the highest frequency). The random selection of a word is not influenced by any other elements such as the frequency count of a specific translation against others; the selection for multiple matched inputs is purely done randomly. This issue causes the BLEU scores to be inconstant due to the fact that for every translation done which are affected with multiple matching inputs, the BLEU scores of the translations vary for a given occurrence. In using frequencies, the answers are static, but the issue also implies to it. For instance, the shortcut "*sn k n?*", given that the frequency count of *sana* (may) is 43, then the translation would be "*sana ka na?*" (may you already) instead of the correct "*saan ka na?*" (where you already). Due to the system's inability to take other adjacent words into context during normalization, the system runs into trouble when it misinterprets words to be translated. There are some cases wherein a given input contains a word that the system sees as a shortcut word even though in context it is already a normalized word. The system searches for a match in the dictionary for every token passed from the input resulting into all tokens passed as candidates for normalization. There are a number of existing words that contain this phenomenon especially when dealing with more than two languages, in the study's case – Filipino including code switching or English. An example of an input that can be misinterpreted is "*San Jose CA*"; the given is considered as a normalized word from a human being's perspective but when the system translates this example, it returns "*saan Jose CA*" wherein it mistakenly substitutes *San* due to the fact that an entry *san - saan* (where) can be found in the word-dictionary. This issue results to a lower BLEU score.

Another issue found during evaluation is that the lack of entries for a dictionary can lead to poor results. This issue is quite difficult to resolve since the dictionary must contain all of the possible variations of a word. There are multiple variants for a single word, especially when dealing with the Filipino language. There are certain shortcut words such as *kaung* which should be normalized to *kayong* but the system was unable to successfully normalize it, even though an obvious entry of *kau* is paired with the normalized word *kayo* (you), because the first pair is not included in the dictionary. Each token passed in the system is normalized as a whole and therefore does not normalize only parts of the word while retaining the others. To fully address the issue, the system is required to fully include all the variations of a word, gaining less possibility in missing a match.

6. CONCLUSIONS AND FUTURE WORK

In this paper, the issue on the inclusion of shortcut texts in data gathering for text processing was addressed through the creation of the system which normalizes the informal texts. Applying the Dictionary Substitution Approach with the word dictionary, the system resulted into having a BLEU score of 0.5780 without the use of frequencies and 0.6077 with frequencies. Furthermore, it can be noted that using modern day and commonly used types of data results to high BLEU scores. Overall, the use of frequency counts on tied multiple matches enable the system to perform better.

Although numerous issues were encountered during the evaluation of the system, the study still finds the DSA approach as flexible. It is still possible to resolve the issues encountered by fixing the dictionaries used, particularly adding entries based on the noted shortcut words that were not normalized. In addition, changing the language domain of the dictionary may turn the scope of the system from Filipino to another language. As the system was also able to normalize half of the data while considering the fact that two languages are taken into consideration: Filipino and English, this shows that DSA in NORM can still perform accordingly even if there are different languages used.

The issues regarding the use of the approach for the Filipino language still exist and are concentrated on to the ambiguity of the shortcuts and the substituted word. For future works wherein the dictionary substitution approach is chosen as a normalization approach, the factors should be considered in building the system: increasing the dictionary size, making the normalization process case sensitive, adding phrases to the translation rules, and automatically adding entries in the dictionary. In using a different approach, the ones that consider contexts is suggested for it may improve a normalization system's output by knowing or having a heuristic on which output should be substituted at a current phrase.

7. ACKNOWLEDGEMENTS

We would like to express our deep gratitude to the following for their continuous support in our study: Prof. Charibeth Cheng, Prof. Leif Syliongka, and Ms. Mica Tiu.

8. REFERENCES

[1] Beburlee. 2013. Definition of Jejemon. *Collins Dictionary*. http://www.collinsdictionary.com/submission/3390/Jejemon/ .

[2] Brill, E., and Moore, R. 2000. An Improved Error Model for Noisy Channel Spelling Correction. *In Proceedings of the 38th Annual Meeting on Association for Computational*

Linguistics, 2000, Stroudsburg, PA, Association for Computational Linguistics, USA, 286-293.

[3] Cabatbat, J. T., and Tapang, G. A. 2013. *Texting Styles and Information Change of SMS Text Messages in Filipino*. Technical Report. World Scientific Publishing Company.

[4] Dita, S., Roxas, R., and Inventado, P. 2009. Building Online Corpora of Philippine Languages. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, Stroudsburg, PA, Association for Computational Linguistics, USA, 646-653.

[5] Liu, F., Liu, Y., Wang, B., and Weng, F. 2011. Insertion, Deletion, or Substitution? Normalizing Text Messages without Pre-categorization nor Supervision. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, Stroudsburg, PA, Association for Computational Linguistics, USA, 71-76.

[6] Papineni, K., Roukos, S., Ward, T., & Zhu, W. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, Stroudsburg, PA, Association for Computational Linguistics, USA, 311-318.

[7] Raghunathan, K., & Krawczyk, S. 2009. *CS224N: Investigating SMS Text Normalization using Statistical Machine Translation*. Department of Computer Science, Stanford University.

[8] Thurlow, C. 2003. Generation Txt? The sociolinguistics of young people's text-messaging. *Department of Communication*, 1, 1, 1-27.

[9] Wannisinghe, J. 2011. Semiotic Analysis of Short Message Service (SMS). *International Journal of Communicology*, 1, 1, 11- 19.