ANALYZING WORD ERROR RATE (WER) OVER PART OF SPEECH (POS) TAGGING

[1]THEINT ZARNI MYINT, [2]MYO KAY KHAING

[1,2]University of Computer Studies, Yangon
E-mail: chutpit@gmail.com, theintzarnimyint@ucsy.edu.mm

Abstract- The task of the part of speech (POS) tagging has been applied by various ways. This paper represents POS tagging by measuring the minimum edit distance between English and Myanmar phonetic based on English language. The input of this task is voice that is automatically generated the text output by android speech recognizer. The proposed system is originally defined as the minimum number of deletions, insertions, and substitutions required for transforming one word into another using Levenshtein Distance (LD) algorithm. This paper presents a framework for extraction of linguistics details from standard words error rates WER over ten basics POS classes. It also carried out a detailed analysis of inflectional errors and missing words error between English and Myanmar phonetic. Tested on the 1000 corpus on English and this system is developing on android platform using eclipse workbench.

Keywords- Word Error Rate (WER), POS Tagging, Speech Recognition.

I. INTRODUCTION

Mobile phones have become an integral part of our Everyday life, causing higher demands for content that can be used on them. Smart phones offer customer enhanced methods to interact with their phones but the most natural way of interaction remains speech. Market for smart mobile phones provides a number of applications with speech recognition implementation. Voice recognition technology is the process of identifying and understanding the voice signals of a user, which is converted into text or commands for a program.

Part-of-speech (POS) tagging is a well-knownproblem and a common step in many naturallanguage processing applications such as rulebased machine translation (RBMT). A POS taggerattempts to assign the correct POS tag or lexicalcategory to all words of a given text, usually byrelying on the assumption that a word can be assigned a single POS tag by looking at the POS tagsof the neighboring words.

In this work the voice recognition technology is applied into a laboratory information system for identifying each user's voice. However, due to the low recognition rate of the current speech recognition systems, in order to increase the accuracy of information retrieval, appropriate post-processing is required.

Therefore a variety of automatic evaluation measures have been studied over the last years. The most widely used are Word Error Rate (WER), Position independent word Error Rate (PER), the BLEU score and the NIST score. These measures have shown to be valuable tools for comparing different systems as well as for evaluating improvements within one system. However, these measures do not give any details about the nature of translation errors. Therefore some more detailed analysis of the generated output is needed in order to identify the main problems and to focus the research efforts. A framework for human error analysis has been proposed in, but as every human evaluation, this is also a time consuming task.

This article shows a framework for calculating WER over POS tagging that is for estimating the contribution of each POS tagging to overall word error rate. Although this work focuses on POS classes, the method can be easily extended to other types of linguistic information. The translation system used is the phrase-based statistical machine translation system. This work presents a framework for extraction of linguistic details from standard word error rates WER use for an automatic error analysis. We present a method standard word error rates WER over basic POS classes. We also carry out a detailed analysis of inflectional errors which has shown that the results obtained by our method correspond to those obtained by a human error analysis. In addition, we propose an approach for analyzing missing word errors.

II. RELATED RESEARCHES

This research combines three main elements: speech recognizer on android, standards words error rate on POS tagging and Levenshtein Distance.

Entering text on a small mobile device is often sluggish and error-prone, compared to text input via full-sized keyboard.
This fact has been noticed by mobile software manufacturers and today many phone operating systems offer text inputby means of automatic speech recognition (ASR). For example, on the Android platform, Google Voice Search (application that lets user search the web using a spoken query) is supported in 29 languages and accents (at the end of January2012).

In the informationtheory andcomputer science, the Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. It is named afterVladimir Levenshtein, who considered this distance in 1965.

The use of linguistic knowledge for improving the performance of a statisticmachine translation system is investigated in many publications for various language pairs (like for example (NieBen and Ney, 2000)(Goldwater and McClosky, 2005)). Its use for the analysis of translation errors is still a rather unexplored area. Some automatic methods for error analysis using base forms and POS tags are proposed in(Popovic et al., 2006; Popovic and Ney, 2006).

These measures are based on difference between WER and PER which are calculated separately over POS class using subsets extracted from the original texts. Standard overall WER and PER of the original texts are not at all taken into account. In this work, the standard WER is decomposed and analyzed using POS tagging and LD algorithm to become the higher accuracy over android speech recognizer.

III. SYSTEM DESIGN

Our approach receives as input produced by the voice and automatic generated by the speech recognizer on android to automatic transform speech to text. This system is adapted to input message in English. But the output is not matched with user speech because English and Myanmar phonetic is not match. For example, when the user told the input is “Shwedagon Pagoda”, the output of speech recognizer is “Sharia Pagoda”. So, this system proposed an approach for obtaining more detail about actual translation errors in the generated output by using the word error rate (WER). It employs the following modules: (1) the standard procedure for evaluating machine translation output is done by comparing the hypothesis document hyp with given reference translations ref, each one consisting of K sentences (or segments); (2) extracting the distribution of missing words over POS classes using WER errors: (3)estimation of inflectional errors by use of WER errors in our application.

A. Speech Recognition on Android

Speech recognition is the process of converting voice signal into corresponding text or commands. In this research ,the user speaks the interesting places in Myanmar through the microphone and the converted text is inserted in the corresponding result field and stored in the database. The speech recognition application in smart phones is incorporated to implement this speech to text conversion operation. Fig. 1 depicts the basic system architecture of the speech recognizer.

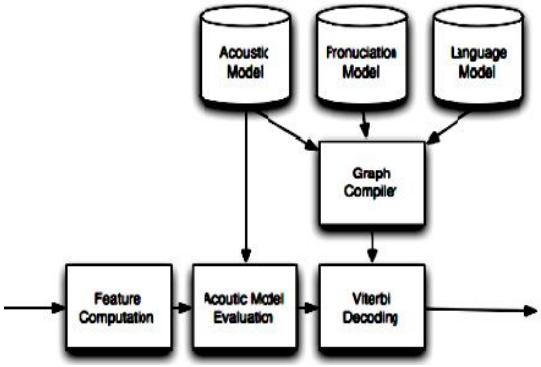

Fig 1. Block Diagram of Speech Recognizer

Speech recognition system can be divided into several blocks: feature extraction, a coustic models database which is built based on the training data, dictionary, language model and the speech recognition algorithm. Analog speech signal must first be sampled on time and amplitude axes, or digitized. Samples of speech signal are analyzed in even intervals. This period is usually 20 ms because signal in this interval is considered stationary. Speech feature extraction involves the formation of equally spaced discrete vectors of speech characteristics. Feature vectors from training database are used to estimate the parameters of acoustic models. Acoustic model describes properties of the basic elements that can be recognized.The basic element can be a phoneme for continuous speech or word for isolated words recognition.Dictionary is used to connect acoustic models with vocabulary words. Language model reduces the number of acceptable word combinations based on the rules of language and statistical information from different texts. Speech recognition systems, based on hidden Markov models are today most widely applied in modern technologies. They use the word or phoneme as a unit for modeling. The model output is hidden probabilistic functions of state and can’t be deterministically specified. State sequence through model is not exactly known. Speech recognition systems generally assume that the speech signal is a realization of some message encoded as a sequence of one or more symbols . To effect the reverse operation of recognizing the underlying symbol sequence given a spoken utterance, the continuous speech waveform is first converted to a sequence of equally spaced discrete parameter vectors.

B. Standards Word Error Rate (WER)

The word error rate (WER) is based on the Levenshtein distance (Levenshtein, 1966) - the minimum number of substitutions, deletions and insertions that have to be performed to convert the generated text hyp into the reference text ref. The

reference document ref consists of R reference translation for each sentence. Let the length of the hypothesis sentence hyp_k be denoted as N_{hyp_k} , and the reference lengths of each sentence N_{ref_k} . Then the hypothesis length of the document is $N_{hyp}=\sum_k N_{hyp_k}$, and the total reference length is $N_{ref}=\sum_k N_{ref_k}$ where N_{refk} is defined as the length of the reference sentence with the lowest sentence-level error rate. The WER of the hypothesis hyp with respect to reference ref is calculated as:

$$WER = \frac{1}{N^*_{ref}} \sum \min_r d_L(ref_{k,r}, hyp_k) \qquad (1)$$

where $d_L(ref_{k,r}, hyp_k)$ is the Levenshtein distance between the reference sentence $ref_{k,r}$ and the hypothesis sentence hyp_k . The calculation is performed using a dynamic programming algorithm.

C. WER decomposition Over POS classes

The dynamic programming algorithm for WER enables a simple and straightforward identification of each erroneous word which actually contributes to WER. Let err_k denote the set of erroneous words in sentence k with respect to the best reference and p be a POS class. Then n (p, err_k) is the number of errors in err_k produced by words with POS class p. It should be noted that for the substitution errors, the POS class of the involved reference word is taken into account. POS tags of the reference words are also used for the deletion errors, and for the insertion errors the P OS class of the hypothesis word is taken. The WER for the word class p can be calculated as:

$$WER(p) = \frac{1}{N^*_{ref}} \sum_{k-1}^{K} n(p, err_k) \qquad (2)$$

The sum over all classes is equal to the standard overall WER. An example of a reference sentence and hypothesis sentence along with the corresponding P OS is shown in Fig. 2.

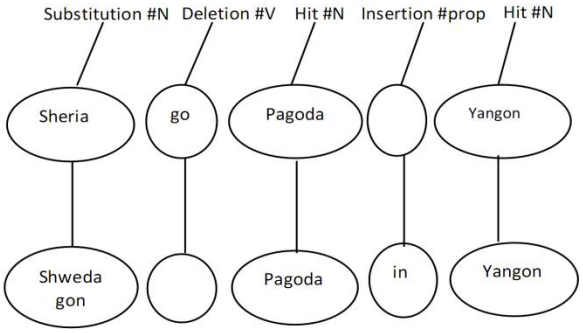

Fig.2. Example for illustration of actual errors : a POS tagged reference sentence and a corresponding hypothesis sentence

The W ER errors, i.e. actual words participating in WER together with their POS classes can be seen in table1. The reference words involved in W ER are denoted as reference errors, and hypothesis errors refer to the hypothesis words participating in WER.

Table I WER errors: actual words which are participating in the word error rate and their corresponding POS tagged

Hypothesis errors	Reference errors	Error type to update
Sheria #N	Shwedagon #N	substitution
Go #V		deletion
	In #prop	insertion

Standard WER of the whole sentence is equal to 3/5=60% .The contribution of nouns is WER (N) = 1/5=20%, of verbs WER (V) =1/5=20% and of proposition WER (prop) =1/5= 20%.

D. Levenshtein distance (LD)

Levenshtein distance (LD) is a measure of the similarity between two strings, the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t.

The greater the Levenshtein distance, the more different thestrings are. In our case, the source string is the input, and the target string is one of the entries in the dictionary.

The Levenshtein algorithm (also called Edit-Distance) calculates the least number of edit operations that are necessary to modify one string to obtain another string. The most common way of calculating this is by the dynamic programming approach.

A matrix is initialized measuring in the (m, n)-cell the Levenshtein distance between the m-character prefix of one with the n-prefix of the other word.

The matrix can be filled from the upper left to the lower right corner. Each jump horizontally or vertically corresponds to an insert or a delete, respectively.

The cost is normally set to 1 for each of the operations. The diagonal jump can cost either one, if the two characters in the row and column do not match or 0, if they do. Each cell always minimizes the cost locally. This way the number in the lower right corner is the Levenshtein distance between both words.

An example of finding LD between the user speech “Shwedagon” and the speech recognition output “should I go” is shown in Fig.2.

		S	h	o	u	l	d		i		g	o
	0	1	2	3	4	5	6	7	8	9	10	11
S	1	0	1	2	3	4	5	6	7	8	9	10
h	2	1	0	1	2	3	4	5	6	7	8	9
w	3	2	1	1	2	3	4	5	6	7	8	9
e	4	3	2	2	2	3	4	5	6	7	8	9
d	5	4	3	3	3	3	3	4	5	6	7	8
a	6	5	4	4	4	4	4	4	5	6	7	8
g	7	6	5	5	5	5	5	5	5	6	6	7
o	8	7	6	5	6	6	6	6	6	6	7	6
n	9	8	7	6	6	7	7	7	7	7	7	7

Fig.3. Example of Levenshtein Distance

According to the above figure, it has the substitutions, deletions, insertions on the minimum edit distance between two corpuses is 7. So the WER of these two sentence is 7/9=0.78 ~ 78%.

IV. RESULTS AND EXPERIMENTS OVER APPLICATION

The decomposition error rate is different types from above section for standard error rates. But for more precise information about certain phenomena some kind of future analysis is required. In this work, we investigate the next two possible aspects for error analysis: estimation of inflectional errors and extracting the distribution of missing words.

A. Inflectional Errors

Inflectional errors can be estimated using WER based on POS tagging classes and based forms. From each reference-hypothesis sentence pair, only erroneous words which have the common based forms are taken into account. English and Myanmar phonetic are more different. The nouns are the most problematic category in both languages.

For example, the user's input is "Shwedagon" that is noun over POS classes and the recognizer output is "Shirago" that is also noun over POS classes.

So, they don't match. To be matched with these, we are using the LD algorithm. Inflectional error rate in this example is present for these nouns, and is calculated in the same way as WER over POS classes, i.e. is WER (p) = 1/1= 1 and WER over LD algorithm is 7/9= 0.78~78%.

B. Missing Words

Distribution of missing words over POS classes can be extracted from WER in the following ways: the words considered as missing are those which occur as deletions in WER errors and at the same time occur as reference. The information about the base forms like proposition (in,for,at) is used in order to eliminate inflectional errors. The number of missing words is extracted for each word class and then normalized over sum of all classes. For example sentence pair presented in Table 1, from the WER errors, the words proposition "in" will be missing words.

V. RESULTS OVER ERROR ANALYSIS

This analysis is based on the android platform. So, these papers focus on the results for the mobile. This system uses the android recognizer that is automatically generated from speech to text. It can apply on the internet accesses. The translation is performed on the English language and the error analysis is done on the users' input and the recognizer output in English. English reference is stored in the domain database that is already created as a early stage. The English hypothesis is performed by the speech recognizer on android.

The input and output fall into the POS tagging approach that is used the Standard ford parser which can generate the POS classes. In this way, all reference and hypotheses are provided with POS tags.

The decomposition of WER is done over the ten main POS classes: noun (N), verbs (C), adjectives (A), adverbs (ADV), pronouns (PRON), determiners (DET), prepositions (PREP), conjunctions (CON), numerals (NUM) and punctuation marks (PUN). Inflectional error rates are also estimated for each POS classes and the distribution of missing words over ten POS classes is estimated using the WER.

A. WER decomposition over POS classes

The decompositions of WER over ten basic POS classes are shown in Fig.4. The largest part of both word error rates comes from the two most word classes, namely nouns and verbs, and the least critical classes are punctuations, conjunctions and numbers.
The number of erroneous nouns and pronouns is higher in English output. Here it will use the LD algorithm to match the users' input.

The WER of adjectives, determiners and prepositions are significantly worse in the Myanmar phonetics. In addition, the decomposed error rates give an idea of where to put efforts for possible improvements of the system. For example, working on implements of verb translations could reduce about 10% WER, working on nouns up to 8% WER, whereas there is no reason to put too much efforts on e.g. adverbs since this could lead only to about 2% of WER reduction.

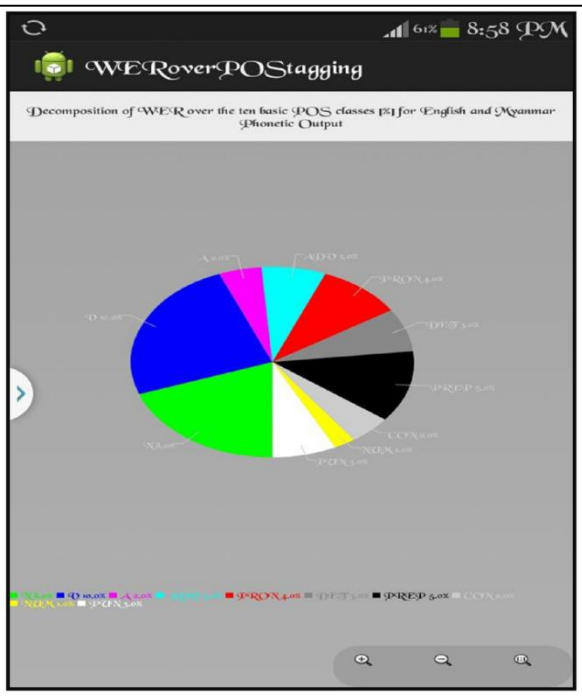

Fig.4. Decomposition of WER over ten basic POS classes for English and Myanmar Phonetics

B. Inflectional Errors over POS classes

Inflectional error rates for ten POS classes are presented in Fig.5. For comparing English and Myanmar phonetics, these are significant only one POS class: nouns. The nouns are the most problematic category. The verbs have not inflectional errors because the English verbs already have been meaning for Myanmar language. Nouns have a higher error rate between English and Myanmar phonetic. So, it can apply only on the nouns that the WER is about 90% and other POS classes are about 10% on WER.

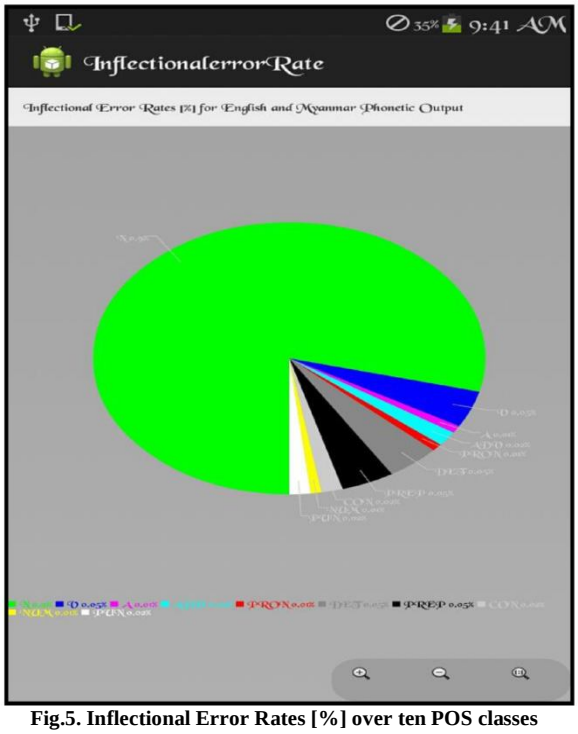

Fig.5. Inflectional Error Rates [%] over ten POS classes

C. Missing Words over POS classes

The distribution of missing words over POS classes is displayed on Fig.6. Most missing words for both languages are proposition. It can be shown in the Fig.1. Verbs are more often missing, as well as determiners. A probablereason is the disproportion of the number of occurrences for those classes between references and hypothesis.

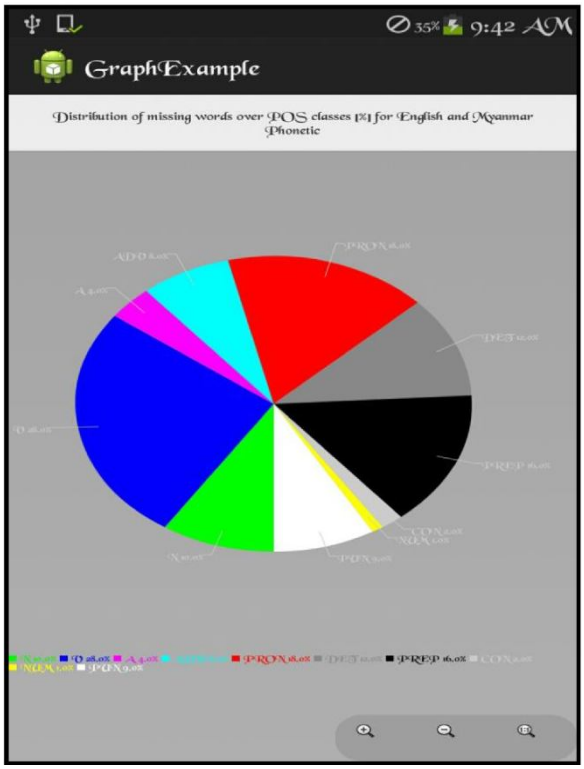

Fig.6. Distribution of missing words over ten POS classes

CONCLUSION

In this paper demonstrate the results of recognition give user option to select the most accurate result using POS tagging. This work is planned to implement the model of speech recognition for matching user speech and the output results. This system is active on the Internet. If the connection is lost, this application will not be used. This work is especially for different kinds of places such as pagoda, hotel, market, etc. in our country. That is why the words are related to our Myanmar phonetics. This work presents a framework for extraction of linguistic details from standard word error rates WER and Leveshtein distance use for automatic error analysis. This paper demonstrated an approach for the decomposition of the standard word error rates, inflectional error analysis, missing words analysis over ten POS classes.

REFERENCES

[1] J. Schalkwyk, D.Beeferman, F. Beaufays,B. Byrne, C.Chelba, M. Cohen, M. Kamvar, and B. Strope, ""Your Word is my Command": Google search by voice: A case study," in Advances in Speech Recognition, Amy Neustein, Ed., pp. 61–90.Springer US, 2010

[2] A. Fujii, K. Itou, and T. Ishikawa, “A method for open-vocabulary speech-driven text retrieval,” in proceeding of the 2002 conference on Empirical Methods in Natural Language Processing, pp.188-195, 2002.

[3] H.Jung, G. Geunbae Lee, W.Choi, K.K. Min, and J. Seo, “Multi-lingual question answering with high portability on relational databases,” IEICE transactions on information and systems, vol E-86D, No2, pp.306-315, 2003..

[4] R. Malouf., “A comparison of algorithms for max-imum entropy parameter estimation”, In proceedings of CoNLL-2002, 49-55, Taipei,Taiwan, 2002.

[5] A.Juhui, L.Seungwoo, and G. G. Lee, “Automatic acquisition of Named Entity tagged corpus from World Wide Web,” in proceedings of the 41st annual meeting of the ACL (poster presentation), 2003.

[6] G. G. Lee, C.Jeongwon, and L.Jong-Hyeok, “Syllable pattern-based unknown morpheme segmentation and estimation for hybrid part-of-speech tagging of Korean,”

[7] D. Kim, H. Lee, C. Seon, H. Kim, and J. Seo, "Speakers' Intention Prediction Using Statistics of Multi-level Features in a Schedule Management Domain," in Proc. ofAssociation for Computational Linguistics on Human Language Technologies, pp. 229-232, 2008.

[8] S. Kang, H. Kim, J. Seo, “A Reliable Multidomain Model for Speech Act Classification,” Pattern Recognition Letters, vol. 31, no 1, pp. 71-74, 2010.

[8] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. Boosted MMI for model and feature-space discriminative training. In Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2008.

[9] [Online]. Available:http://dx.doi.org/ 10.1016/ j.patrec. 2010.09.029 C. Seon, H. Kim, J. Seo, “Efficient Appointment Information Extraction from Messages in MobileDevices with Limited Hardware Resources,” Pattern Recognition Letters, vol. 32, no 2, pp. 127-133, 2011.

[10] [Online]. Available: http://dx.doi.org/ 10.5626/ JCSE.2011.5.2.141 M. Lee, D. Han, “Ubiscript: A Script Language for Ubiquitous Environment,” Journal of Computing Science and Engineering, vol. 5, no 2, pp. 141–149, 2011.

[11] M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. strope. Deploying GOOG-411: Early lessons in data, measurement, and testing. In Proceedings of ICASSP, pages 5260{5263, 2008.

★★★