USING WEB TEXT TO IMPROVE KEYWORD SPOTTING IN SPEECH

Ankur Gandhe, Long Qin, Florian Metze, Alexander Rudnicky, Ian Lane

Language Technologies Institute
Carnegie Mellon University
{ankurgan,lqin,fmetze,air,ianlane}@cs.cmu.edu

Matthias Eck

Mobile Technologies
Moffet Field, CA; U.S.A.
matthias.eck@jibbigo.com

ABSTRACT

For low resource languages, collecting sufficient training data to build acoustic and language models is time consuming and often expensive. But large amounts of text data, such as online newspapers, web forums or online encyclopedias, usually exist for languages that have a large population of native speakers. This text data can be easily collected from the web and then used to both expand the recognizer's vocabulary and improve the language model. One challenge, however, is normalizing and filtering the web data for a specific task. In this paper, we investigate the use of online text resources to improve the performance of speech recognition specifically for the task of keyword spotting. For the five languages provided in the base period of the IARPA BABEL project, we automatically collected text data from the web using only LimitedLP resources. We then compared two methods for filtering the web data, one based on perplexity ranking and the other based on out-of-vocabulary (OOV) word detection. By integrating the web text into our systems, we observed significant improvements in keyword spotting accuracy for four out of the five languages. The best approach obtained an improvement in actual term weighted value (ATWV) of 0.0424 compared to a baseline system trained only on LimitedLP resources. On average, ATWV was improved by 0.0243 across five languages.

Index Terms— keyword spotting, low resource, data filtering, language modeling, web text

1. INTRODUCTION

To perform speech recognition for a given task and language, three models are required: an acoustic model (AM), a language model (LM) and a pronunciation lexicon. The performance of these models depends heavily on the amount of data used during training. Furthermore, the training data is often collected manually, which is very costly and time consuming. On the other hand, with the advent of the Internet, there are now large amounts of easily accessible data online, in particular the text data. To effectively use these resources, however, we must accurately identify which resources and documents best match a given task, then collect and filter that data. The resulting data is used to both improve the language model and expand the recognizer's vocabulary.

There has been significant prior work describing techniques to collect text data from the web to improve the recognition accuracy. Such approach is frequently used when building a speech recognition system for a specific task on common languages, such as English or Mandarin [1, 2], which have strong presence on the Internet. Earlier work [3] updated the language model dynamically by querying recently recognized words for similar web text. [4] showed improvements in word error rate (WER) by obtaining counts for n-grams that were infrequent or not observed in the original training data from the web. More recent approaches have focused on collecting matched genre, i.e. tesxt in conversational-style, from the web. [1] showed that by using small amounts of in-domain data and n-grams as queries, one could retrieve text similar to the in-domain data. A similar approach was studied in [2] for Mandarin. [5] decomposed n-grams into classes to allow different interpolations for content words and function words.

The most common way of integrating the collected web data into the language model is to build separate language models for the seed corpus and web data then combine these models using standard interpolation techniques [6]. [7] also proposed building topic specific language models by classifying crawled documents into piles of background, topic and rejected data. Normally, before using the web text, relevant portions of the text need to be selected. The majority of work in this area has been on ranking the external text according to the similarity with the in-domain text and discarding low ranking chunks [8, 9, 10]. [8] defined similarity as the perplexity of the sentence measured by the in-domain language model. Another measurement proposed by [10] uses relative perplexity of in-domain and external language models. Other work on text selection makes use of part-of-speech and language specific properties [11].

In this paper, we investigate improving the keyword spotting performance for low resource languages as provided in the IARPA BABEL program. Rather than only using data provided under the program, we introduce additional text resources automatically collected from the web into our sys-

tems. For all languages, we begin with a seed corpus consisting of the LimitedLP resources provided by BABEL. This corpus consists of 10 hours of conversational speech with transcripts (approx.$\sim$100 k tokens) and pronunciation lexicon for five languages (Cantonese, Pashto, Tagalog, Turkish and Vietnamese[1]). All these languages are relatively rare on the Internet and most of the web text is in formal written form. As a result, it is very challenging to find text data online to match the conversational training data of those languages. Furthermore, under the LimitedLP condition, the out-of-vocabulary (OOV) rate is very high, up to 13%, which is a significant issue for the keyword spotting task. Therefore, we investigate methods to automatically collect and filter text data from online resources that most closely match in-domain data in terms of language usage and style for building expanded keyword spotting systems. We introduce three methods for collecting text data from different online sources, and propose two approaches for data filtering: one inspired by the perplexity similarity ranking work of [10] and the other is an extension of OOV word detection and recovery [12]. Our experimental results showed that data filtering approaches are better than directly using all the crawled web data in five BABEL languages. Furthermore, by incorporating the filtered web data for language modeling, we achieved significant improvement in keyword spotting over the baseline LimitedLP system.

2. WEB CRAWLING

We collected the external data from WWW using three different methods and data sources. Since the five BABEL languages are rare and do not have particularly strong and known presence on the web, we used the following three methods: crawling websites returned by web search queries, extracting text from Wikipedia XML dumps, and direct crawling of some known news websites.

2.1. Web search queries

The web search queries were unigrams and bigrams with $4 <$ frequency < 50 extracted from the LimitedLP training data. These frequency limits were chosen empirically and resulted in the best results without using too many queries. The number of queries per language was in the range of 2,000-5,000. We used Google to retrieve the results, from which the top 30 results were retained and crawled. The advantage of this method is that the collected text will contain n-grams found in the training data. However, the style and domain of the crawled text can be different from the conversational style of the training data.

[1]This effort uses the IARPA Babel Program language collection release babel101-v0.4c, babel104-v0.4Y, babel105-v0.4-rc, babel106-v0.2f, babel107-v0.7

2.2. Wikipedia dumps

From Wikipedia, the XML dump for each BABEL language was downloaded and all text was extracted. The clear and enforced structure makes Wikipedia easy to parse. This data also provides a high vocabulary coverage. However many terms in Wikipedia tend to be too esoteric or technical for the conversation training data, and also have the problem of very different style of writing.

2.3. Crawling news websites

For Vietnamese, an additional set of Vietnamese news websites were crawled. This is a large dataset of well structured data, but it is not targeted towards the vocabulary of the training data, as this data was crawled completely unsupervised.

3. DATA FILTERING

We compared two methods for filtering the crawled text data. In one method, we performed perplexity filtering by ranking crawled text according to the in-domain perplexity and selecting sentences with low perplexity. In the other method, we used all the filtered text but the recognition vocabulary is expanded through OOV word detection, where we added possible OOV words in the development and testing speech into the LimitedLP vocabulary. Then, we trained our language model using the crawled text and the expanded vocabulary.

3.1. Filtering through perplexity ranking

We ranked the web data according to its similarity and consistency to the conversation transcriptions and then used the top ranking sentences for language modeling. We applied a unigram language model built from the conversation training data to calculate the perplexity of each sentence in the crawled text data. This measurement produced lower perplexities for sentences that only contained words in the vocabulary and some sentences with a small fraction of unseen words. Since we also want to lower the OOV rate, we used the following scoring function as our final ranking metric

$$S_i = \lambda \cdot PPL_{LIM}(sent_i) - (1-\lambda) \cdot \sum_{w_j \in OOV, sent_i} PPL_{WEB}(w_j), \quad (1)$$

where S_i is the score assigned to a sentence 'i' (lower is better), PPL_{LIM} and PPL_{WEB} are the perplexities produced by language models trained on the LimitedLP data and web data respectively. The idea is to introduce unseen words to lower the overall OOV rate while matching the style of training text and not introducing too many unknown words. To ensure coherence of added text, we kept λ values very high for all languages.

3.2. Filtering through OOV word detection

Another way of filtering the crawled text data is to train the language model from in-vocabulary (IV) words that appear in the LimitedLP vocabulary of the conversation training data and OOV words appear in the development and testing speech. Specifically, we considered words present in the crawled data but not in the LimitedLP vocabulary as OOV words, from which we trained the sub-lexical units and built a hybrid recognition system. Then, we performed hybrid decoding to detect OOV words in the development and testing speech. After that, we searched the crawled data for words which have similar pronunciations as the detected OOV words, and added those words into the LimitedLP vocabulary to build a filtered vocabulary. Finally, we re-trained a new LM from the crawled text data, with the vocabulary LimitedLP to the filtered vocabulary.

3.2.1. OOV word detection using the hybrid system

In our hybrid system, we applied a hybrid lexicon and hybrid language model during decoding to detect the presence of OOV words. We considered the decoded sub-lexical units in the hybrid system output as detected OOV words. And we collected a list of OOV words together with their pronunciations from the hybrid decoding result of the development and testing speech.

The hybrid lexicon was obtained by integrating sub-lexical units and their pronunciations into the IV word lexicon. First, the pronunciations of all OOV words in the crawled data were estimated through the grapheme-to-phoneme (G2P) conversion [13], and then used to train the sub-lexical units. After that, OOV words in the crawled text were replaced by corresponding sub-lexical units to get a new hybrid text corpus. Finally, a hybrid language model was trained from this hybrid text data. When training the hybrid language model, sometimes two or more OOV words might appear consecutively in the crawled data. Hence, after representing OOV words using sub-lexical units, the word boundary between two OOV words was lost. To solve this problem, we added two more symbols into the sub-lexical sequence of each OOV word, which are the word start "^" and word end "$". We also used those symbols to segment a sequence of sub-lexical units in the hybrid system output into multiple detected OOV words.

To achieve good G2P conversion performance, we trained a 6-gram joint sequence model with short length graphone units as suggested in [14]. We also assigned an OOV cost C_{OOV} to control how likely the system would be to decode OOV words during decoding. By tuning C_{OOV} on the development data, we could find the optimal configuration of our system to achieve the target OOV detection performance. More details of our hybrid system can be found in [12].

3.2.2. Filtering the crawled data

After detecting OOV words in the development and testing speech, we searched words in the crawled data to find which ones have similar pronunciation as the detected OOV words. Then, we added those similar sounding words into the LimitedLP vocabulary to construct a filtered vocabulary. Finally, we built a new recognizer using the filtered vocabulary and performed a second pass decoding on the testing speech. This idea is similar to [15], where detected OOV city names were incorporated into the system for a second pass decoding.

We measured the phonetic similarity between detected OOV words and words in the crawled data by computing the distance between their pronunciations. Specifically, the phonetic distance $d(o, w)$ between the detected OOV word o and word w in the crawled data was formulated as the normalized edit distance between their pronunciations p_o and p_w:

$$d(o,w) = \frac{edit(p_o, p_w)}{|p_o| + |p_w|} \tag{2}$$

where $|p_o|$ and $|p_w|$ are the number of phones in p_o and p_w. Since the pronunciations of OOV words were collected from the hybrid decoding result, they might incorporate recognition errors. Particularly, some similar phones were more easy to be mis-recognized than the other phones. Therefore, we adopted a modified edit distance that compensates for the acoustic confusability between phones [16-19],

$$\begin{aligned} edit(0,0) &= 0 \\ edit(i,0) &= i \\ edit(0,j) &= j \\ edit(i,j) &= min \begin{cases} edit(i-1,j)+1 \\ edit(i,j-1)+1 \\ edit(i-1,i-1)+c(i,j). \end{cases} \end{aligned} \tag{3}$$

In Eq. 3, $c(i, j)$ is the confusability between phones at positions i and j

$$c(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1 - p(i,j) & \text{if } i \neq j, \end{cases} \tag{4}$$

where $p(i, j)$ is the probability of mis-recognizing two phones, which was estimated from the recognition result of the conversation training speech.

The pronunciation of OOV words in the crawled web data was estimated from G2P conversion, which might also introduce errors. When we selected words from the crawled data for augmenting the LimitedLP vocabulary, we did not only add words whose $d(o, w)$ is zero. Instead, if $d(o, w)$ was smaller than a threshold, we then incorporated w into the LimitedLP vocabulary. The threshold was tuned on the development data to cover as many OOV words as possible but also maintain a fixed of the filtered vocabulary.

	Cantonese		Pashto		Tagalog		Turkish		Vietnamese	
	Token	Vocab	Token	Vocab	Token	Vocab	Token	Vocab	Token	Vocab
LimitedLP Only	110k	6k	113k	6k	72k	6k	71k	10k	125k	3k
No Filtering	1966k	43k	2880k	97k	3180k	160k	33450k	955k	17105k	37k
Perplexity Filtering	1966k	43k	841k	34k	1480k	63k	1591k	31k	4936k	13k
OOV Filtering	1966k	35k	841k	34k	1480k	57k	1591k	31k	4936k	12k
FullLP Data	871k	19k	881k	18k	584k	21k	556k	38k	985k	6k

Table 1: The number of tokens of the LM training data and the vocabulary size of recognizers built from different data.

	Cantonese		Pashto		Tagalog		Turkish		Vietnamese	
	OOV	PPL	OOV	PPL	OOV	PPL	OOV	PPL	OOV	PPL
LimitedLP Only	4.9	90.7	4.4	170.2	7.7	123.3	12.5	290.2	1.6	160.3
No Filtering	3.3	142.5	1.7	209.4	4.3	164.2	4.0	577.4	0.5	158.8
Perplexity Filtering	3.3	142.5	2.3	194.1	4.8	153.8	8.8	363.9	0.6	158.7
OOV Filtering	3.4	140.9	2.3	194.6	4.9	154.5	9.3	353.1	0.7	158.2
FullLP Data	2.1	87.4	1.4	154.5	3.0	125.3	5.2	330.8	0.3	125.8

Table 2: The OOV rate (in %) and perplexity of recognizers built from different data.

4. EXPERIMENT SETUP

4.1. Data

The primary goal of this paper is to investigate the use of the external web text for keyword spotting in very low resource conditions. We performed experiments on the five BABEL languages (Cantonese, Pashto, Tagalog, Turkish and Vietnamese), in which 10 h of telephone conversation data is available. The number of tokens of the LM training data and the vocabulary size of recognizers built from different data are given in Table 1. We present the statistics of : the baseline LimitedLP condition where only the transcription of the 10-hour speech was used, the No Filtering conditions where all the crawled web data was used, the Perplexity and OOV filtering conditions where only the filtered text data was used, and finally a FullLP data condition where additional 90 hours of conversation text data available in the BABEL program was used. It can be seen that the language model training data is very small under the LimitedLP condition but very large in the no filtering condition. The vocabulary sizes are much smaller in the filtering conditions than that in the no filtering condition, as we filtered out many irrelevant words through either perplexity ranking or OOV word detection. To produce comparable results, the web text used to train the language model in OOV filtering was the same as produced by perplexity filtering. Only the vocabulary was modified using the OOV filtering technique.

4.2. System building

After crawling and cleaning the web text data, we first estimated the pronunciation for OOV words in it. For Pashto, Tagalog, Turkish and Vietnamese, the pronunciation of OOV words was estimated through the grapheme to phoneme (G2P) conversion. The G2P model was built only from IV words under the LimitedLP condition. For Cantonese, we trained a character-to-phoneme (C2P) mapping to produce pronunciations for OOV words. Because of that, if there was an unknown characters in an OOV word, we were unable to estimate the pronunciation for the word, hence we ignored that word.

We used the SRILM toolkit for language modeling [20]. All language models were trigram models with modified Kneser-Ney smoothing. When using the crawled data, we first trained a language model from it, then interpolated it with the in-domain language model. The interpolation weight was tuned on the development data using the "best-mix" script in the SRILM toolkit. The OOV rate and perplexity of each language under different conditions is shown in Table 2. We find that compared to the LimitedLP-Data condition, the OOV is reduced by more than 50% in most languages when using all the crawled text (No-Filtering). However, the perplexity is much higher due to the increase in vocabulary size. By filtering the crawled data, the perplexity of the final language model can be lowered at the expense of slightly increased the OOV rate. It is important to keep the perplexity of models reasonable to avoid deviation from the conversation style of the given LimitedLP data.

5. RESULTS

5.1. The keyword spotting results

To evaluate the proposed crawling and filtering techniques, we decoded the 10 h development data of the five BABEL languages using language models built from different data and vocabularies. We then indexed the decoding results and searched about 300-1000 keywords in each language.

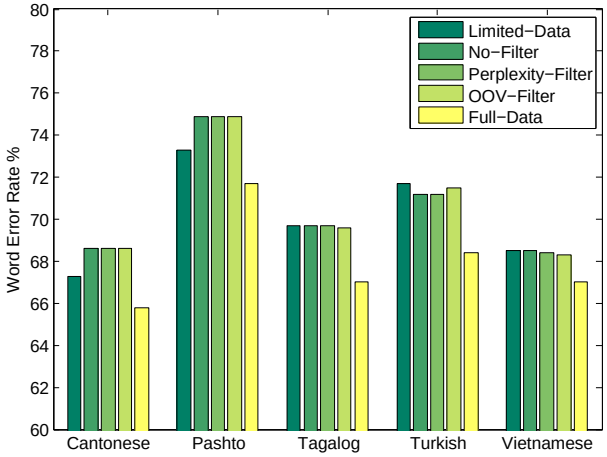

Fig. 1. The recognition performance of recognizers built from different data.

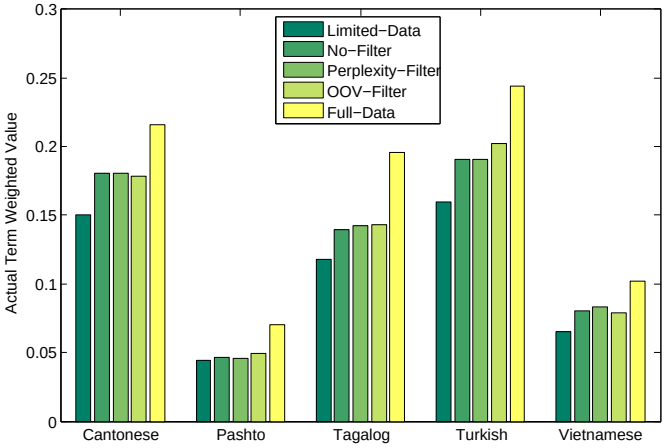

Fig. 2. The keyword spotting performance of systems built from different data.

The word error rate (WER) or character error rate (CER, for Cantonese) of recognizers built from different data is shown in Figure 1. We can find that the WER slightly decreases when building recognizers with the crawled web data in Turkish, it remains the same in Tagalog and Vietnamese, but increases a little bit in Cantonese and Pashto. Because of using extra in-domain text data, the WER of the FullLP Data system, which is built on the transcription of the 100-hour conversational speech, is the lowest. Within three systems using the crawled web data, the recognition performance does not change much, indicating that filtering the crawled data doesn't hurt the recognition performance.

We used the actual term weighted value (ATWV) to measure the keyword spotting performance. As shown in Fig. 2, different from the observation of WER, the keyword spotting performance is significantly improved when using the crawled web data, as more OOV words are recognized and retrieved by our system. Between the "No-Filtering" system which uses all the crawled web data and the "Perplexity-Filtering" or "OOV-Filtering" system which uses only filtered data, the ATWV score of the latter system is better in four out of five languages. The most gain was obtained in Turkish, where the LimitedLP system has a rather high OOV rate.

Between "Perplexity-Filtering" and "OOV-Filtering", the performance highly depends on the kind of language and the OOV word detection accuracy. In Pashto, Tagalog and Turkish, where we had better OOV word detection performance, as presented in Table 3, the "OOV-Filtering" approach can further boost the ATWV score. But in languages where OOV word detection often fails , the "Perplexity-Filtering" is more preferable.

5.2. The OOV word detection results

The OOV word detection performance determines the quality of the filtered vocabulary. For example, if we only found very few OOV words from the development and testing speech, then we will probably get no improvement by integrating them into the vocabulary. On the other hand, even if we could detect many OOV words in the development and testing speech, we might not be able to filter out a lot of irrelevant crawled data, especially if many of them were false alarms produced by our system. Therefore, we used the F1 score which measures both precision and recall to evaluate the OOV word detection performance,

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \quad (5)$$

	Cant.	Pash.	Taga.	Turk.	Viet.
Precision	0.18	0.17	0.26	0.34	0.07
Recall	0.79	0.63	0.55	0.63	0.50
F1	0.29	0.26	0.35	0.45	0.13

Table 3: The OOV word detection performance on the development data of each language.

From Table 3, we can find the OOV word detection performance of the hybrid system on the development data of each language. It can be seen that we have better OOV word detection performance in Turkish and Tagalog, slightly worse performance in Cantonese and Pashto, and the lowest F1 score in Vietnamese. By comparing with Table 2, we can find that the OOV word detection performance highly correlates with the OOV rate of a language. This may because we learned sub-lexical units from OOV words in the crawled web data. If the OOV rate is low, we already have most words of a language in our vocabulary, then we do not have many OOV tokens in the crawled data for training the sub-lexical units and hybrid system. Furthermore, with low OOV rate, the hybrid system usually report a large number of false alarm errors.

6. CONCLUSIONS

In this paper, we evaluated the performance of language models built on the crawled web data for keyword spotting. We collected the web text data from different sources using different methods, and compared several approaches to filter this data. Our experimental results on five languages in the IARPA BABEL program in the LimitedLP condition showed that the keyword spotting performance was substantially improved when integrating the web text data into our systems. Furthermore, pre-filtering the crawled web data yielded further improvement in four out of five languages. On average, the ATWV score increased by 0.0243 across five languages.

7. ACKNOWLEDGEMENTS

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ ARL, or the U.S. Government.

8. REFERENCES

[1] I. Bulyko and M. Ostendorf, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," *Proc. HLT-NAACL-2003*, vol. 2, pp. 7-9, 2003.

[2] T. Ng, M. Ostendorf, M. Y. Hwang, M. H. Siu, I. Bulyko and L. Xin, "Web-data augmented language models for Mandarin conversational speech recognition," *Proc. ICASSP-2005*, pp. 589-592, 2005.

[3] A. Berger and R. Miller, "Just-in-time language modelling," *Proc. ICASSP-1998*, vol. 2, pp. 705-708, 1998.

[4] X. J. Zhu and R. Rosenfeld, "Improving trigram language modeling with the World Wide Web," *Proc. ICASSP-2001*, vol. 1, pp. 533-536, 2001.

[5] R. Iyer, M. Ostendorf and H. Gish, "Using out-of-domain data to improve in-domain language models," *IEEE Signal Processing Letters*, vol. 4, no. 8, pp. 221-223, 1997.

[6] F. Liu et al., "IBM Switchboard progress and evaluation site report," *Proc. LVCSR Workshop*, Gaithersburg, MD, 1995.

[7] A. Sethy, P. G. Georgiou and S. Narayanan, "Building topic specific language models from webdata using competitive models," *Proc. Interspeech-2005*, pp. 1293-1296, 2005.

[8] I. Bulyko, M. Ostendorf, M. H. Siu, T. Ng, A. Stolcke and Ö. Çetin, "Web resources for language modeling in conversational speech recognition," *ACM Trans. Speech Lang. Process.*, 2007.

[9] R. Sarikaya, A. Gravano and Y. Q. Gao, "Rapid language model development using external resources for new spoken dialog domains," *Proc. ICASSP-2005*, vol. 1, pp. 573-576, 2005.

[10] K. Weilhammer, M. N. Stuttle and S. Young, "Bootstrapping language models for dialogue systems," *Proc. ICSLP-2006*, 2006.

[11] S. Schwarm and M. Ostendorf, "Text normalization with varied data sources for conversational speech language modeling," *Proc. ICASSP-2002*, vol. 1, pp. 789-792, 2002.

[12] L. Qin and A. Rudnicky, "OOV word detection using hybrid models with mixed types of fragments," *Proc. Interspeech-2012*, 2012.

[13] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434-451, 2008.

[14] S. F. Chen, "Conditional and joint models of grapheme-to-phoneme conversion," *Proc. Eurospeech-2003*, pp. 2033-2036, 2003.

[15] O. Scharenborg and S. Seneff, "A two-pass strategy for handling OOVs in a large vocabulary recognition task," *Proc. Interspeech-2005*, pp. 1669-1672, 2005.

[16] R. Wagner and M. Fischer, "The string-to-string correction problem," *J. ACM*, vol. 21, no. 1, pp. 168-173, 1974.

[17] K. Audhkhasi and A. Verma, "Keyword search using modified minimum edit distance measure," *Proc. ICASSP-2007*, vol. 4, pp. 929-932, 2007.

[18] M. Pucher, A. Turk, J. Ajmera and N. Fecher, "Phonetic distance measures for speech recognition vocabulary and grammar optimization," *Proc. the 3rd Congress of the Alps Adria Acoustics Association*, 2007.

[19] H. Printz and P. Olsen, "Theory and practice of acoustic confusability," *Computer Speech & Language*, vol. 16, pp. 131-164, 2002.

[20] A. Stolcke, "SRILM - an extensible language modeling toolkit," *In Proc. ICSLP-2002*, pp. 901-904, 2002.