AUTOMATIC KEYWORD SELECTION FOR KEYWORD SEARCH DEVELOPMENT AND TUNING

Jia Cui[1]*, Jonathan Mamou*[2]*, Brian Kingsbury*[1]*, and Bhuvana Ramabhadran*[1]

[1]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
[2]IBM Haifa Research Labs, Haifa 31905, Israel

ABSTRACT

In this paper, we investigate the problem of automatically selecting textual keywords for keyword search development and tuning on audio data for any language. Briefly, the method samples candidate keywords in the training data while trying to match a set of target marginal distributions for keyword features such as keyword frequency in the training or development audio, keyword length, frequency of out-of-vocabulary words, and TF-IDF scores. The method is evaluated on four IARPA Babel program base period languages. We show the use of the automatically selected keywords for the keyword search system development and tuning. We show also that search performance is improved by tuning the decision threshold on the automatically selected keywords.

Index Terms— spoken term detection, keyword search, keyword selection, query selection

1. INTRODUCTION

All search tasks, whether it is searching through vast quantities of spoken archives, locating names, places, events, identifying terms similar to ones spoken before, searching based on examples, or precisely pinpointing information, have different user models and metrics associated with them [1, 2]. The Babel [3] keyword search task is to find all of the occurrences of a "keyword," a sequence of one or more words presented in the target language's orthography, in a corpus of un-segmented speech data. Ideally, an automatic process for selecting keywords for keyword search development would incorporate a user model. However, an accurate model of human search behavior is not yet available because keyword search in audio is still a developing technology, and few user studies have been performed to provide data for developing such models. Moreover, we expect that the selection of keywords by a user may be idiosyncratic and highly task dependent. Thus, we must resort to statistical methods in which the goal is to sample a set of development keywords that are somehow similar to a collection of human-selected keywords for the metric and task at hand. The approach used by the Babel program's testing and evaluation team to create development and evaluation keywords is described in [4]. A variety of manual and automatic methods is used to identify candidate keywords and phrases; then automatically selected candidate lists are vetted and glossed by native speakers. The final keyword set is selected to achieve a target distribution across a variety of factors: keyword length, keyword frequency in the test set and types of keywords and phrases. In the keyword evaluation procedure, a keyword search system finds all possible occurrences of a set of target keywords in an audio collection, assigning a confidence score to each detected instance. Then, a decision threshold is applied to label hits that are deemed to be true occurrences. Adjusting the decision threshold (denoted θ) permits a user to trade off between two kinds of errors: misses and false alarms. Setting a higher threshold decreases the probability of erroneously detecting a keyword ($P_{\text{FA}}(\theta)$) and increases the probability of missing a keyword ($P_{\text{miss}}(\theta)$), while setting a lower threshold has the opposite effect. The set of $(P_{\text{FA}}(\theta), P_{\text{miss}}(\theta))$ pairs produced by sweeping through a large range of decision thresholds is displayed as a detection error trade-off (DET) curve [5].

In the Babel program, the evaluation metric is term-weighted value (TWV): a measure that summarizes system performance for a specific assignment of costs to misses and false alarms [6, 4]. We report results in terms of two different measures: **ATWV**, the actual term-weighted value, which is the TWV achieved with a pre-specified decision threshold; and **MTWV**, the maximum term-weighted value, which is the TWV achieved at the optimal setting of the decision threshold.

We are interested in selecting automatically a set of development keywords for any language in order to be able to tune the decision threshold and to reduce the gap between ATWV and MTWV. What kind of keywords will provide similar optimal thresholds as a set of evaluation keywords? In order to select such keywords, we need to identify properties of keyword groups that would cover a wide range of user models and will impact keyword search performance.

The rest of the paper is organized as follows. First, we briefly introduce the Babel data sets for the base period languages and the experimental setup (Section 2). We describe the features which may affect the keyword search performance, with supporting statistical analysis (Section 3). We propose a fast, computationally practical, keyword selection method with the features described in Section 3 (Section 4). We summarize the results while highlighting key observations (Section 5). We relate these methodologies to prior work (Section 6). Finally, we conclude (Section 7).

2. DATA SETS AND EXPERIMENTAL SETUP

Results are reported on four IARPA Babel program base period language collections: Pashto (release babel104b-v0.4bY), Turkish (release babel105b-v0.4), Tagalog (release babel106b-v0.2g), and Vietnamese (release babel107b-v0.7). The data collection covers a broad selection of speaker dialects and ages, is gender-balanced, and is collected from a wide variety of environments over multiple telephone networks and through many different handsets. For each language, three different data sets are used: (1) the *training set* that contains data for ASR models training; (2) the *development set* that contains audio data with its manual transcription and the associated keyword list, for tuning the systems; and (3) the *evalpart1 set* that contains a subset of the evaluation audio data and the associated keyword list. The development and evaluation data comprises telephone conver-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

Language	Keyword Type	MTWV
Pashto	IV	0.4183
	OOV	0.1370
Turkish	IV	0.5570
	OOV	0.2323
Tagalog	IV	0.5667
	OOV	0.1283
Vietnamese	IV	0.3996
	OOV	0.3564

Table 1. MTWV performance as a function of the keyword type on evalpart1 data set.

sations, while the training data is a mixture of conversational and scripted material. Note that for Pashto, Turkish and Tagalog, the development keyword lists were generated by Babel performers, while for Vietnamese, the development keyword list was generated by the Babel program testing and evaluation team. For each language, the training data can be used in two ways representing two different amounts of transcribed material: (1) Full Language Pack (FullLP), consisting of 20 hours of word-transcribed scripted speech, 80 hours of word-transcribed conversational telephone speech, and a pronunciation lexicon; and (2) Limited Language Pack (LimitedLP), consisting of a 10-hour subset of FullLP plus the remaining audio without transcription.

The ASR system is described in more details in [7]. Briefly, we have used a speaker-adapted deep neural network hybrid model [8, 9] with discriminative pre-training, frame-level cross-entropy training and state-level minimum Bayes risk sequence training. The models were built with the IBM Attila toolkit [10]. A 3-gram LM with modified Kneser-Ney smoothing [11] is applied.

The keyword search system is implemented using the OpenFst toolkit [12] and is described in [13, 7, 14]. Scores are normalized between 0 and 1 using Sum-to-One Normalization as explained in [15]. ATWV and MTWV are evaluated using the F4DE NIST Evaluation tool [16]. The lattices were generated with IBM SA DNN acoustic models [17].

3. FEATURE SELECTION

What kind of features or properties affect the keyword search performance or what kind of keywords are easy to search? Some answers are intuitive. For example, a long keyword which we have seen in the training data frequently is easy to search. This section will give both statistical and intuitive reasons for feature selection.

3.1. Out-of-Vocabulary

Table 1 presents the keyword search performance as a function of the keyword type (in-vocabulary or out-of-vocabulary). It shows that for all the languages, keywords which include only in-vocabulary words have in general better keyword search performance than the keywords containing out-of-vocabulary (OOV) words. Since the keyword search is based on speech recognition results, this observation is also intuitively justified. We call this feature **oov** showing whether the keyword contains word which is OOV.

Language	N-gram order	MTWV
Pashto	1	0.3589
	2	0.4795
	3	0.5990
Turkish	1	0.4810
	2	0.7337
	3	0.8543
Tagalog	1	0.4885
	2	0.6763
	3	0.7220
Vietnamese	1	0.0798
	2	0.3602
	3	0.5256
	4	0.5726

Table 2. MTWV performance as a function of the length of the keyword in words (N-gram order) on evalpart1 data set.

Language	# characters	MTWV
Pashto	4	0.2744
	5	0.3670
	6	0.4440
	7	0.5141
Turkish	4	0.2834
	5	0.4235
	6	0.5060
	7	0.5400
Tagalog	5	0.3760
	6	0.4384
	7	0.5443
	8	0.5839
	9	0.6312
Vietnamese	6	0.3367
	7	0.3669
	8	0.4415

Table 3. MTWV performance as a function of the length of the keyword in characters on evalpart1 data set.

3.2. Keyword Length

Table 2 and Table 3 present the keyword search results for keywords with different lengths, measured by count of words and count of characters respectively. We observe that the longer the keyword, the better the keyword search performance. Note that some languages have shorter words like Vietnamese, which has average of 3.5 characters in a word, while some languages have longer words like Turkish, with an average of 17.2 characters in a word. Also, there are 79% unigram keywords in Turkish and 4% unigram keywords in Vietnamese. For our task, we are interested in finding language-dependent features since the ultimate goal of Babel program is to be able to deal with ANY language. So we add two more features: count of syllables (**sylLen**) and count of phones (**phoneLen**), as well the the count of words (**ngram**) in the keywords. Word segmentation for these languages was provided to us as part of the training material.

count in Train	MTWV
0	0.3733
1	0.5120
2-9	0.4269
>= 10	0.2437

Table 4. MTWV performance as a function of keyword count in training data on Vietnamese development query terms.

3.3. Additional Features

Table 4 presents the search results for keywords with different counts in the training data for Vietnamese Full LP[1]. A clear gap in keyword search performance is seen among different groups of keywords. Part of this difference can be attributed to keyword lengths as discussed in Section 3.2. Intuitively, this feature does seem to carry complementary information. We denote this feature **trainFrq**.

In addition to the above features derived from a detailed analysis of keyword search performance results, we propose three additional features:

1. **Keyword frequency in the development data set (devFreq)**: very often, the development data used in a real-world application, reflects a word or topic distribution that is closer to the evaluation or current distribution. Therefore, it's important to include distributions from the development data in the automatic keyword selection process.
2. **Term Frequency Inverse Document Frequency (TF-IDF)**: in real-world applications, keywords are often named entities, such as names, locations, events or actions. Given that we have no information about the unknown and low-resource language, we can have a rough idea if a keyword is a very common expression or a meaningful keyword by computing its TF-IDF value in each conversation.
3. **Scripted code from training data (scriptcode)**: another resource that we can take advantage of, is the scripted data in the training corpus. Each scripted file has a code indicating whether the content words are numbers, digits, locations, names etc. Since each word in the keyword might occur in multiple scripted codes (file), we take the most frequent script code associated with this keyword as a feature. The value is null if there is no script file associated with the keyword.

4. KEYWORD SELECTION METHOD

Our goal is to match the feature distribution of given keywords (also called target keywords) and candidate keywords. When there are no human-generated queries are given, this method could use statistics of known query terms from other languages as targets. It could also be used to produce a larger set of development queries than can be reasonably produced by people. In order to do this, we sample candidate n-gram keywords for $n = 1..N$ from both training data and development data. The selection program samples a set of keywords from the candidates. The optimal case is that for each feature, the marginal distribution of target and selected keywords are the same. However, it is a combinatorial problem and it is computationally expensive. So we take a sub-optimal strategy to satisfy the joint distribution of features instead of the marginal distribution. We have shown in Section 3 eight different features. Some feature values are numerical, such as, frequencies and TF-IDF values. Such feature values are quantized into 5 different bins. Nevertheless, data sparseness is still an issue. For most joint distributions, there will not be enough candidates to sample from. In such cases, we back-off, sampling candidates to satisfy only partially joint distributions. Given the back-off strategy, sampling features in a specific order may matter for some languages, although we did not observe this in the languages we considered.

[1]We conducted this experiments only on Vietnamese FULL LP data.

5. EXPERIMENTS

5.1. Correctness

The following experiment tests the algorithm correctness. We use the 200 given Vietnamese development keywords as the target and we sample another set of n-grams as a new set of keywords. We compare their DET curves in Figure 1 (a). There is no overlap between the target keywords and the automatically selected keywords. The green curve with filled circles is the DET curve obtained with the given development keywords. The lowest magenta curve (in bold) with filled triangles is the DET curve obtained on 4000 randomly sampled keywords. Most of the keywords are included in the development data, therefore, the random selection attempts to match the distribution of the keywords in the development data. Note that the DET curve of evalpart1 data is the blue one with filled squares. The DET curve of randomly selected keywords is further away from the evalpart1 curve than the DET curve of development keywords. The remaining two cyan and red curves are generated by the automatic selection program with different feature orders, each containing 4000 keywords. "Sample 4K" uses the following feature order **ngram, phoneLen, sylLen, oov, trainFrq, devFrq, scriptcode, TF-IDF** while "reorder sample 4K" uses **oov, sylLen, devFrq, tf-idf, scriptcode, ngram, phoneLen, trainFrq**. The latter yields a performance closest to the keyword set in evalpart1 and hence was used in the remaining experiments. Compared to randomly selected keywords, using the given features has yielded DET curves which are closer to those obtained from evalpart1 keywords.

5.2. Keyword Search

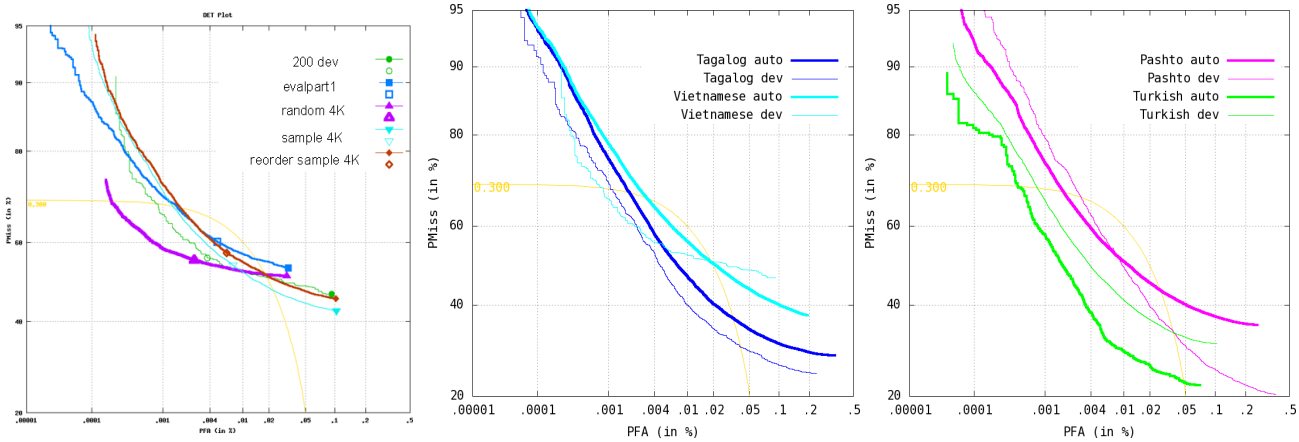
Next, we conducted keywords selection across languages, using evalpart1 keywords from multiple languages. We rotate the experiments among the different Babel languages. For all the languages, we tuned the decision threshold either on the performer-generated development keywords or on our automatically selected development keywords, and measured ATWV on the evalpart1 data set for each language, and compare to MTWV and the optimal threshold on the evalpart1 data set. The results of these tests are summarized in Table 5 for both full and limited language packs. For Turkish, Tagalog and Vietnamese, we see better keyword search performance from tuning on the automatically selected keywords. Figure 1 (b and c) show the DET curves of automatically selected keywords vs. development keywords for all the four languages. We observe that the DET curves are relatively close.

6. RELATED WORK

Query expansion, in which a seed query is reformulated to improve recall, is widely used in Information Retrieval [18]; however, such methods are not applicable to this problem, where no seed queries are available. NIST developed a term selection tool [19] that randomly selects keywords (unigrams, bigrams and trigrams) based on

Language	Language Pack	WER%	evalpart1		performer		automatic	
			MTWV	opt. thresh.	ATWV	thresh.	ATWV	thresh.
Pashto	FullLP	52.1	0.4191	0.016	0.4184	0.016	0.4177	0.020
Turkish	FullLP	49.7	0.5558	0.017	0.5503	0.021	0.5541	0.018
	LimitedLP	65.0	0.2590	0.028	0.2578	0.027	0.2528	0.017
Tagalog	FullLP	48.3	0.5276	0.020	0.5232	0.014	0.5267	0.018
Vietnamese	FullLP	55.9	0.3499	0.004	0.3341	0.008	0.3485	0.004
	LimitedLP	69.3	0.1689	0.006	0.1658	0.007	0.1687	0.006

Table 5. Keyword search system performance on evalpart1 data set achieved by tuning on performer-produced development or automatically generated development keywords.



(a) Keyword selection with Vietnamese development queries: randomly selected v.s. algorithm selected

(b) DET curves comparison for development queries and auto-selected queries for Vietnamese and Tagalog

(c) DET curves comparison for development queries and auto-selected queries for Pashto and Turkish

Fig. 1. Compare DET curves of auto-selected query terms and those of the development and evaluation query terms on different languages

input transcriptions. The method we describe here produces better lists because it more closely matches the characteristics of human-generated keyword lists. The most closely related work of which we are aware is the development by BBN [20], also in the context of the Babel program, of a keyword selection tool that attempts to match the distributions of keyword features to a reference distribution, where the features include keyword frequency, keyword length (in phonemes), and an acoustic confusability measure.

7. CONCLUSION

We have presented an automatic procedure for selecting lists of keywords to be used in the development and tuning of keyword search systems. By approximately matching the marginal distribution of features of the keywords in the selected list to a target distribution (which may be derived from manually selected keywords for another language), we are able to select development keywords that behave similarly to manually selected keywords. Crucial features to match include the number of OOV keywords; the distribution of keyword lengths, measured in terms of syllables; and the frequency of the keywords in the development set. Experiments on four different languages show that these automatically selected lists can be used to set the detection threshold properly: on three of the four test languages, the detection threshold optimized on the automatically selected keywords gives better performance than a detection threshold set on manually selected keywords. As a future work, we plan to apply this automatic keyword selection procedure on option period 1 Babel languages (given that no development keyword list will be provided in the framework of the program). We plan also to use these keywords for learning the keyword search scoring and system combination functions.

8. ACKNOWLEDGMENT

We are grateful to Janice Kim of IBM Research for providing software support for the keyword search toolkit. This effort uses the IARPA Babel Program language collections IARPA-babel104b-v0.4bY (Pashto), IARPA-babel105b-v0.4 (Turkish), IARPA-babel106-v0.2g (Tagalog), and IARPA-babel107b-v0.7 (Vietnamese). 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-0012. 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.

9. REFERENCES

[1] Dagobert Soergel, "Indexing and retrieval performance: The logical evidence," *JASIS*, vol. 45, no. 8, pp. 589–599, 1994.

[2] D. W. Oard, D. Soergel, D. Doermann, X. Huang, G. C. Murray, J. Wang, B. Ramabhadran, M. Franz, S. Gustman, J. Mayfield, et al., "Building an information retrieval test collection for spontaneous conversational speech," in *ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004, pp. 41–48.

[3] M. Harper, "IARPA Solicitation IARPA-BAA-11-02," `http://www.iarpa.gov/solicitations_babel.html`, 2011.

[4] "OpenKWS13 Keyword Search Evaluation Plan," `http://www.nist.gov/itl/iad/mig/upload/OpenKWS13-EvalPlan.pdf`, 2013.

[5] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech*, 1997, pp. 1895–1898.

[6] J.G. Fiscus, J. Ajot, J.S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proceedings of ACM SIGIR Workshop on Searching Spontaneous Conversational*. Citeseer, 2007, pp. 51–55.

[7] B. Kingsbury, J. Cui, X. Cui, M. J. F. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. C. Woodland, "A high-performance Cantonese keyword search system," in *Proc. ICASSP*, 2013.

[8] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Improving training time of deep belief networks through hybrid pre-training and larger batch sizes," in *Proc. NIPS Workshop on Log-linear Models*, 2012.

[9] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. Interspeech*, 2012.

[10] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEEWorkshop on Spoken Language Technology*, 2010.

[11] S. F. Chen and J. T. Goodman, "An empirical study of smoothing techniques for language modeling," in *In Proceedings of the 34th ACL*, 1996, pp. 310–318.

[12] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," *Implementation and Application of Automata*, pp. 11–23, 2007.

[13] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *Proc. ASRU*, 2009, pp. 404–409.

[14] M. Saraclar, A. Sethy, B. Ramabhadran, L. Mangu, J. Cui, X. Cui, B. Kingsbury, and J. Mamou, "An empirical study of confusion modeling in keyword search for low-resource languages," in *Proc. ASRU*, 2013.

[15] J. Mamou, J. Cui, X. Cui, M. J. F. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. C. Woodland, "System combination and score normalization for spoken term detection," in *Proc. ICASSP*, 2013.

[16] "NIST Tools," `http://www.itl.nist.gov/iad/mig/tools/`.

[17] J. Cui, X. Cui, B. Ramabhadran, J. Kim, B. Kingsbury, J. Mamou, L. Mangu, M. Picheny, T. N. Sainath, and A. Sethy, "Developing speech recognition systems for corpus indexing under the IARPA Babel program," in *Proc. ICASSP*, 2013.

[18] S. E. Robertson, "On term selection for query expansion," *J. Doc.*, vol. 46, no. 4, pp. 359–364, Jan. 1991.

[19] "NIST Term Selection Tool," `www.itl.nist.gov/iad/mig//tests/std/tools/TermSelectionTools.20061020.tgz`.

[20] "Email communication from Rich Schwartz and Damianos Karakos about the pdf matching method used for generating artificial keywords for the Babel project.," .