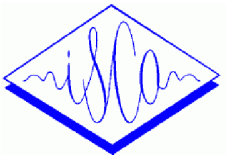
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

# White Listing and Score Normalization for Keyword Spotting of Noisy Speech

*Bing Zhang, Richard Schwartz, Stavros Tsakalidis, Long Nguyen, Spyros Matsoukas*

Raytheon BBN Technologies, Cambridge, MA, USA
{bzhang,schwartz,stsakali,ln,smatsouk}@bbn.com

## Abstract

We present a method that avoids the problem of a large vocabulary recognition system missing keywords due to pruning errors or degraded speech. The method, called white listing, assures that all tokens of all of the keywords are found by the recognizer, albeit with a low score. We show that this method far outperforms methods that attempt to increase recall by using subword models. In addition, we introduce a simple score normalization technique based on mapping the decoding score for a keyword to the probability of false alarm for that keyword. This method has the advantage that it can be estimated for all keywords with reliability, even though there might not be any examples of those keywords in the training or tuning set. This makes the scores of all keywords consistent at all ranges, which allows us to use a single consistent score for all keywords. We show that this method reduces the average miss rate by about a factor of 2 for the same false alarm rate. The method can also be used for combining multiple keyword spotting systems.

**Index Terms**: keyword search, noise robustness, white list, score normalization

## 1. Introduction

High performance keyword systems typically use large vocabulary recognition to find keywords or key phrases.[1] However, the 1-best answer from a recognizer may often miss a keyword, even though the recognizer has a good score for that keyword. There are two typical reasons for this: 1) the score for a keyword might be lower than that of another word in the vocabulary, 2) the keyword might not be in the recognition vocabulary. For our application, we ignore the second problem because the keywords are known before speech is acquired. The 1-best output of the recognizer provides one extreme of the tradeoff between misses and false alarms. Although the miss rate may not be as low as we would like, the false alarm rate for a recognizer is extremely low. That is, although a large fraction of the words might be misrecognized, the probability that any of them are accidentally recognized as one of the keywords is extremely small. For example, if a recognizer has a word error rate (WER) of 50%, then the nominal probability that any of those recognition errors happens to output one of the keywords is roughly 1/vocabulary size. So if the vocabulary size is 50,000 words, the probability is about $10^{-5}$.

So typically, in order to decrease the miss rate, we also look for hits for the keywords in the recognition lattice, which provides many alternatives. This decreases the miss rate at the cost of more false alarms. However, when presented with severely degraded speech, the miss rate is often still quite high because the word may have been pruned out of the search or the output lattice may not have been deep enough. The miss rate can easily exceed 50%. We can increase the beam width in the search and increase the depth of the lattice, but this can quickly increase the computation and memory to an unacceptable point.

We propose a simple solution to this problem here, which is to inform the recognizer of the set of keywords and "protect" those keywords from being pruned out so that they almost always be in the lattice if they were actually spoken. We call this list of keywords a "white list" in that these words are (almost) always accepted, although possibly with a low score.

A second problem in keyword spotting is related to the scores that different keywords get in a speech recognizer. If we are only looking for a single keyword, the relation between the score and the actual probability that the keyword is present is not important, because we can simply sort the keyword hits according to their score to create a ranked list. However, if we actually want to look for several keywords, the scores for the different keywords must be on a common scale. Thus, many evaluation metrics require that all keywords be given a consistent score such that all keywords can be put on a single ranked list in order that a single threshold could be set for all keywords.

Many keyword search (KWS) systems use an estimate of the posterior probability of the keyword computed from the lattice or n-best answers. However, we observe that this estimate tends to have a bias such that the same "posterior" probability results in different performance for different keywords. So the challenge is to estimate a consistent score, given that we might have very few - if any - samples of the keyword in order to calibrate the measure across keywords.

We propose to estimate the probability of false alarm (pFA) for each keyword as a function of its score. Given any corpus of transcribed speech that is acoustically like the targeted speech domain, we can estimate the mapping between the posterior or usual confidence score and pFA simply by running the keyword spotter on the transcribed speech, finding many false alarms, sorting them by score, and mapping the score to the empirical pFA on the data. The result is a highly accurate estimate of pFA for each keyword that we can use as the new score for the keyword and which will be consistent across all keywords - even though the keywords may not be present in transcribed speech.

In section 2, we survey different approaches for decreasing the miss rate (at the expense of increased false alarm) for keyword spotting. In section 3, we describe the white listing method in detail, along with some of the problems. In section 4, we describe the pFA score normalization approach. And in section 5, we present experimental results showing the benefit of these methods.

## 2. Background

Many studies of speech recognition in noise have been performed, but most of them deal with speech with artificially

[1] We use "keywords" from now on to indicate keywords or key phrases.

added noise, or with speech that is already noisy. In this case, as part of the DARPA RATS (Robust Automatic Transcription of Speech) project. we are dealing with clean speech that has been degraded by being transmitted through 8 different degraded radio channels [1].

The resulting speech varies widely in quality and intelligibility, with various distortions, dropouts, frequency shifts, push-to-talk noise, etc. The speech varies from somewhat intelligible to barely intelligible. The original speech was taken from the Levantine Fisher Corpus [2] which was produced at LDC by having different native speakers of Levantine Arabic speak on the telephone about different topics.

Table 1 shows the effect of this noise on the WER of a recognizer trained and tested under different conditions. Since we actually have the original clean speech available, we can compare the effect of noise on the training or decoding results.

The test sets are created based on the Dev04 test set from the DAPRA EARS program. We selected 200 unique keywords with at least 3 syllables that ocurrs between 1 to 4 times. Dev04r is the corresponding retransmitted set. The training sets are:

- **T1:** 170 hours of the Levantine Fisher Conversational Telephone Speech (LFCTS)
- **T2:** 30 hours of recovered retransmission of an LFCTS subset over 8 degraded radio channels
- **T3:** 190 hours of retransmission of another LFCTS subset

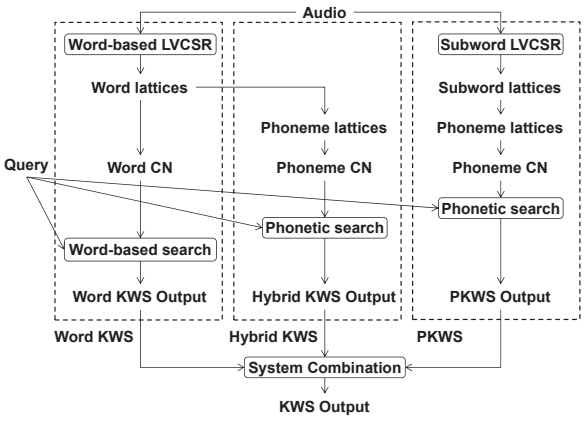



Figure 1: Baseline KWS system diagram

The KWS system is based on [3, 4], which combines word-based and sub-word based systems, as illustrated in Figure 1. The BBN CTS speech recognizer [5] was employed to generate lattices from audio data. The acoustic model is a quinphone cross-word SCTM model trained under the MPE criterion [6] together with speaker adaptation. The LM is an unpruned trigram model.

For some KWS applications, the keywords may not be known at the time the speech recognition is actually performed. In this case, there is a possibility that the keywords may not be in the vocabulary. However, in this project, the keywords are known before the speech is acquired so there is no OOV problem, although the keywords might not be in the acoustic or language model training data. However, we can still miss words entirely due to the severe degradation; the correct keywords are often missing from the recognizer lattice output.

A common solution to the OOV problem is to use a phonetic recognizer instead of a large-vocabulary word recognizer, and then to look for keywords as sequences through a phoneme lattice. This solution can also be used to increase the number of hits for the keywords in order to decrease the miss rate. However, phonetic recognition accuracy is quite poor compared to word recognition - especially on highly degraded speech. At BBN, we have two variations on methods for sub-word models: the "PKWS" (phonetic KWS) system and the "hybrid" system.

In the PKWS system, we build a recognizer that contains a relatively small number of recognition units. The units include single phones, many phone-pairs and syllables, as well as the most common 1000 words [7]. This recognizer can run very quickly but has higher accuracy than a model that just uses single phonemes. The recognizer outputs a lattice of recognized units, which is converted into a lattice of phonemes.

In addition, we know that a whole-word large vocabulary system has high accuracy - not only for the words in its vocabulary, but for the phoneme sequences it recognizes. So we run a normal LVCSR system which outputs a word lattice, and then we convert this word lattice into a second phoneme lattice which can be searched for new words. It is referred to as the "hybrid" system.

Given a keyword, we can search for hits for the word in the output word lattice. However, we can also search for the word in the phoneme lattice using a robust matching procedure that allows phones to be missing. The score for a word match against the phoneme lattice depends on the number of phones that match, the acoustic scores for those phonemes, and several other measures. Different confidence estimation methods have been explored in [8, 9]. We employ a GLM based confidence estimation as in [3]. Thus, depending on the score that we allow, we can find virtually any keyword almost anywhere that at least one of its phonemes was recognized. Finally, we can combine the outputs of these three systems to get a ranked list of scores for all of the keywords using linear combination [10].

| Training Data | Dev04 (clean) | | | Dev04r (noisy) | | |
|---|---|---|---|---|---|---|
| | WER | pMiss | | WER | pMiss | |
| | | IV | OOV | | IV | OOV |
| T1 (clean) | 44.6 | 2.6 | 3.1 | 75.0 | 53.7 | 53.8 |
| T1+T2 | - | - | - | 70.4 | 28.5 | 30.5 |
| T3 | - | - | - | 67.7 | 22.3 | 27.4 |

Table 1: pMiss at 4% pFA on Dev04 and Dev04r using the combined word, hybrid and PKWS system

As we can see, the WER increases dramatically when the model trained on clean speech is tested on degraded speech. The degradation is reduced significantly when the recognizer is trained on degraded speech, but is still significantly worse than the performance on clean speech.

For KWS performance, we chose an operating point with a very high false alarm rate (pFA) in order to emphasize how much the system misses with noisy speech. The false alarm rate is defined as $\frac{\#.falsealarms}{\#.totalwords \times \#.keywords}$, so 4% pFA means 4 false alarms per keyword for every 100 words spoken, which is a very high number. As can be seen, even at this operating point, the pMiss value is still quite high on noisy speech.

We can also measure the recall of the KWS system as the percentage of all true keywords that are output at all with any score - however low. For example, for the 1-best output, this recall was only about 44% on channel A (one of the 8 channels) for the word-based sub-system in T3. For the lattice output, the recall increased, but still was only about 60% - thus 40% of the keywords had no chance of being found.

## 3. KWS White List

Given that we know the set of keywords ahead of time, we can avoid the OOV problem by simply adding the words to the recognizer vocabulary. In order to avoid pruning errors, we implemented a "white list" feature in the recognizer. Simply stated, we provide the list of keywords. During normal beam pruning, we compare the score at a state with the score of the best hypothesis at that instant and remove the state from consideration if its score is below some threshold relative to that highest score (the beam). In this case, if the state belongs to one of the words in the white list, the threshold is much lower (a wider beam) so that it becomes very unlikely for this keyword to be pruned out when it is in fact in the speech. This modification is made to all of the places in the decoder where any pruning takes place. Even though the computation for the keywords is increased by a large factor, the overall computation does not increase by much, since we might only have 200 keywords. The output lattice then contains many hypotheses for the keywords, even though the overall lattice is not much larger.

Although the basic idea is simple, there are many details that must be handled correctly. For example, many of the keywords are actually phrases. Our solution for these is simply to add the whole phrases (as if they were single words) to the recognition vocabulary. This does cause a small problem in that the individual words in a phrase can compete with the whole phrase, thus making the posterior probabilities not strictly correct. We explain how this problem is dealt with in the next section.

Of course, the scores for these additional tokens of the keywords are often quite low, but they are all available for insertion into a ranked list. Using this method, we were able to increase the recall of the word-based system to 98%. In contrast with the various sub-word systems, here we are still using models of words and the language model, which are typically much more powerful than sub-word models.

## 4. Score Normalization

One challenge in KWS is to produce a score for each hit that is consistent across all keywords. That is, the same score will result in the same performance across all keywords. The problem is that the acoustic and language model scores vary across different words due to having different amounts of training, different word lengths and many other factors. As we said before, it is likely that there are very few - if any - tokens of the keywords in a development set. So it is not possible to measure the typical score of a true hit of this keyword.

One common solution is to try to estimate the confidence in a recognized word using many features that include the posterior probability along with these various other features, combined in a classifier, such as a neural network. However, this model is, at best, an approximation of the actual relationship between the score for a particular word and its actual probability of being correct.

However, although we do not have many true tokens of the word, we can generate as many false alarms as we like by running the KWS system on labeled speech and noting all false alarms. We sort the hits for each keyword by their score from best to worst. Then, we compute the probability of false alarm (pFA) for each position in the list as the rank of that answer divided by the total number of words in the corpus. Once we get to the 10th position, this pFA value is fairly well estimated. So now, when running the KWS system on new speech, when we get a score for a hit of a keyword, we can simply look up the pFA that we would expect if we use this score as the threshold. Note that this is not exactly the same as the probability that this individual hit is a false alarm, but it functions in the same way.

Since we can generate a large number of false alarm hits for each keyword, we have reliable estimates for relationship for pFA vs. the score. This means all of the keywords can have a consistent score. In fact, we use the score 1-pFA so that higher scores are better.

A further benefit of this approach relates to the problem related to the competition between phrases and the individual words in the phrase. Let's assume the individual words are not in the white list (the argument still follows if they are). So for this compound keyword, when the score is high, we also expect to have hypotheses for its component words in the lattice, which will decrease the posterior for this compound keyword. However, the pFA normalization will adjust for that directly. When the score for the compound keyword is relatively low, the individual words probably will not be in the lattice, so the problem may not occur and the score normalization map will reflect that as well. So, although this does not completely remove the problem of this artificial competition between compound words and their component words, it largely alleviates the problem implicitly.

Finally, this score normalization method has another application. If you have several different KWS systems that you would like to combine, we typically combine the scores from different systems with some weights. This requires that the scores from the different systems are all similar types of scores, which is not always true. Those weights are estimated on the average to maximize performance. An alternate method is to apply the pFA normalization method to the output of each system. Thus, for each system and each keyword, there will be a map from its score to the likely pFA. We can then combine these pFA values with weighted interpolation as usual, but now the combination is actually keyword-specific.

## 5. Experimental Results

In this section we will show experimental results of the above techniques on the Levantine KWS task in RATS. The main test set is the official development set for phase-1 (dev-1) created in the RATS project by retransmitting clean conversational telephone data over 8 noisy channels, which imitate different pairing of radio transmitters and receivers. It consists of 219 unique keywords in 1.1 hours of noisy speech, with majorities of keywords being multi-word keywords. The total number of occurrences of the keywords is 401.

The acoustic training corpus used in the experiments was created in the same way as the test data. Two incremental releases have been made, which we will refer to as R1 (LDC2011E86) and R2 (LDC2011E111) in experimental setups below. The amount of data for all channels, including the clean channel, is 196 hours and 290 hours for R1 and R2, respectively. The LM training corpus consists of the acoustic training transcript from R1 and R2, which amounts to 504K words, as well as Levantine STT transcript and Levantine web text from the GALE project, which amounts to about 1M words. The baseline KWS system has been described in section 2.

### 5.1. White List Results

In this experiment, we show the effect of using keyword white list by comparing the white-listed system to the baseline (Table

| Expt | Target | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | pMiss @ 0.2% pFA | | | | pFA @ 30% pMiss | | | |
| | Word | HYB | PKWS | Sys-Comb | Word | HYB | PKWS | Sys-Comb |
| Baseline | 45.4† | 40.4 | 55.1 | 38.4 | 0.04‡ | 0.83 | 1.77 | 0.74 |
| White list | **33.7** | 40.4 | 55.1 | 31.2 | **0.27** | 0.83 | 1.77 | 0.26 |

Table 2: KWS results on dev-1 using the keyword white list.
† *@ 0.04% pFA.* ‡ *@ 45.4% pMiss.*

2). The system was trained using release R1.

The word-based KWS system in the baseline could not reach the target operating points, because the deletion rate is too high on the noisy data. In the table we are showing the lowest pMiss and the highest pFA for that sub-system. The hybrid and PKWS systems could, but they generated too many false alarms in order to make the target, due to the problems we pointed out earlier. In contrast, the word-based system with white list not only did significantly better than the individual sub-systems in the baseline, but also much better than the baseline system combination.

### 5.2. Score Normalization Results

Table 3 shows the effect of pFA-based score normalization on dev-1 when the score normalizer is trained on the same set. The improvement is very large compared to the baseline. Although it was a cheating experiment, we believed that the gain should hold even under the fair condition, because the score mapping for each keyword was trained using hundreds to thousands of false alarms, so it should be reliable. In order to verify the claim, we created two development sets of about the same size of dev-1 by holding out some training data from release R2 for development. We then trained the score normalizer using test set "Tune" and applied it to test set "Validation". In Table 4, the first row is the results without pFA-norm; the second row is the results with the cheating setup; the final row is the fair results. These results show that pFA-norm works very well under the fair condition.

| Score Norm | Target | |
|---|---|---|
| | pMiss @ 0.2% pFA | pFA @ 30% pMiss |
| None | 30.0 | 0.20 |
| pFA | **25.2** | **0.10** |

Table 3: KWS results on dev-1 using pFA-norm. Score normalizer is trained on the same set.

| Score Norm Training Set | Test Set | Target | |
|---|---|---|---|
| | | pMiss @ 0.2% pFA | pFA @ 30% pMiss |
| None | Validation | 37.0 | 0.39 |
| Validation | Validation | 34.1 | 0.26 |
| Tune | Validation | **33.2** | **0.28** |

Table 4: Fair score normalization results using Tune and Validation

## 6. Conclusions

We have shown that the white listing method allows us to use whole-word large vocabulary recognition to obtain low miss rates for keyword spotting, even for highly degraded speech. The resulting system alleviates the problem of high miss rate that typical LVCSR systems have with degraded speech, but get much better performance than sub-word solutions.

The pFA score normalization makes the relationship between the scores and performance consistent across all keywords, even if those keywords do not occur in the development data. This is shown to improve the overall performance when the KWS performance measure or the application requires that the scores are consistent.

## 7. Acknowledgement

This paper is based upon work supported by the DARPA RATS Program. It has been approved for public release and distribution is unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

## 8. References

[1] D. Graff, S. Sessa, S. Strassel, and K. Walker, "RATS Data Plan," Linguistic Data Consortium, Tech. Rep., Feb 2011, http://www.ldc.upenn.edu/RATS.

[2] M. Maamouri *et al.*, "LDC2006S29, Arabic CTS Levantine QT training data set 5," Linguistic Data Consortium, Feb 2006.

[3] I. Bulyko, O. Kimball, M.-H. Siu, J. Herrero, and D. Blum, "Detection of unseen words in conversational Mandarin," in *Proceedings of ICASSP*, Kyoto, Japan, Mar 2012.

[4] I. Bulyko, J. Herrero, C. Mihelich, and O. Kimball, "Subword speech recognition for detection of unseen words," in *Proceedings of Interspeech*, Portland, OR, Sep 2012, submitted.

[5] S. Matsoukas *et al.*, "Advances in transcription of broadcast news and conversational telephone speech within the combined EARS BBN/LIMSI system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1541–1556, Sep 2006.

[6] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proceedings of ICASSP*, 2002.

[7] I. Szoke, L. Burget, J. Cernock, and M. Fapo, "Sub-word modeling of out of vocabulary words in spoken term detection," in *IEEE Workshop on Spoken Language Technology*, India, 2008.

[8] T. Mertens, D. Schneider, and J. Kohler, "Merging search spaces for subword spoken term detection," in *Proceedings of InterSpeech*, Brighton, UK, Sep 2009, pp. 2127–2130.

[9] C. W. Han, S. J. Kang, C. M. Lee, and N. S. Kim, "Phone mismatch penalty matrices for two-stage keyword spotting via multi-pass phone recognizer," in *Proceedings of InterSpeech*, Makuhari, Chiba, Japan, Sep 2010, pp. 202–205.

[10] J. Tejedor, D. Wang, S. King, J. Frankel, and J. Colas, "A posterior probability-based system hybridization and combination for spoken term detection," in *Proceedings of InterSpeech*, Brighton, UK, Sep 2009.