An In-Depth Comparison of Keyword Specific Thresholding and Sum-to-One Score Normalization

Yun Wang and Florian Metze

Language Technologies Institute, Carnegie Mellon University
Pittsburgh, PA, U.S.A.
yunwang@cs.cmu.edu, fmetze@cs.cmu.edu

Abstract

The quality of a spoken term detection (STD) system critically depends on the choice of a "thresholding" function, which is used to determine whether to output a candidate detection or not based on its score. In the context of the IARPA Babel program and the NIST OpenKWS evaluation series, the penalty for missing an occurrence depends on the frequency of the keyword, so it is desirable either to apply different thresholds to different keywords, or to normalize the scores before applying a global threshold. This paper compares two widely used thresholding algorithms: keyword specific thresholding (KST) and sum-to-one score normalization (STO), analyzes the difference in their performance in detail, and recommends the use of the "estimated KST" algorithm.

Index Terms: Spoken term detection, IARPA Babel, NIST OpenKWS evaluation, keyword specific thresholding, score normalization

1. Introduction

Spoken term detection is the task of detecting occurrences of text queries (also called *keywords*) in an audio corpus. The pipeline of a typical spoken term detection is shown in Fig. 1. First, the audio data is processed with a speech recognizer. Instead of outputting the single best hypothesis, the recognizer produces multiple hypotheses in the form of lattices or confusion networks [1], which we call the *index*. While the index in our experiments consists of words, it may also consist of morphemes or phones. Next, the index is searched for occurrences of keywords, and each detection is assigned a *raw score* between 0 and 1 based on the posterior probabilities of words in the index. These detections make up the *raw detection list*. Finally, in the "thresholding" step, the score of each detection is compared with a threshold. Detections with scores above the threshold are retained and make up the *final detection list*, which is the output of the system.

The performance of a spoken term detection is evaluated by how many hits, misses and false alarms there are in the final detection list. Several evaluation metrics exist that combine these numbers in different ways, such as the F_1 score, *figure of merit* (FOM) [2], and *actual term weighted value* (ATWV) [3]. In this paper, we deal with the ATWV metric, which is the primary evaluation metric in the IARPA Babel program and the NIST Open Keyword Search (OpenKWS) evaluation series [3].

ATWV is defined as follows:

$$\text{ATWV} = \frac{1}{N}\sum_{w}\left[\frac{N_{\text{hit}}(w)}{N_{\text{true}}(w)} - \beta\cdot\frac{N_{\text{FA}}(w)}{T - N_{\text{true}}(w)}\right] \quad (1)$$

where N is the total number of keywords (excluding those that never occur in the corpus), w stands for any keyword, $N_{\text{true}}(w)$, $N_{\text{hit}}(w)$, and $N_{\text{FA}}(w)$ are the numbers of true occurrences, hits, and false alarms of the keyword w, respectively. Two other constants are involved in the formula: β is a factor that controls the balance between misses and false alarms; in the OpenKWS evaluation, it is set to 999.9. T is the total duration of the audio corpus in seconds, which is around 36,000.

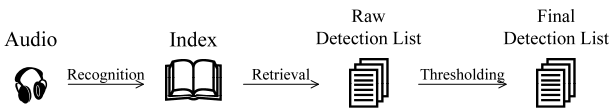

Figure 1: *Pipeline of a spoken term detection system*

If we plug the two constants into Eq. (1), and consider the fact that usually $T \gg N_{\text{true}}(w)$, we can get an intuitive approximation of ATWV:

$$\text{ATWV} \approx \frac{1}{N}\sum_{w}\left[\frac{N_{\text{hit}}(w)}{N_{\text{true}}(w)} - \frac{N_{\text{FA}}(w)}{36}\right] \quad (2)$$

We see that the penalty for a false alarm is almost constant ($1/36N$), while the penalty for a miss depends on the number of true occurrences, or *frequency*, of the keyword. Missing a rare keyword is more costly than missing a frequent keyword.

The fact that the penalty for misses varies with the keyword frequency motivates us to set the threshold separately for each keyword: we should set lower thresholds for rare keywords to avoid misses, and higher thresholds for frequent keywords to avoid false alarms. Alternatively, we may normalize the raw scores so that the detection scores of rare keywords get boosted, and those of frequent keywords get suppressed.

Two representative thresholding algorithms have emerged along the two lines of thinking: keyword specific thresholding (KST) [4] and sum-to-one score normalization (STO) [5]. As we have observed, there is some confusion among researchers about the procedure and relative performance of the two algorithms. The authors of [5], who proposed the STO algorithm, claimed that "[KST] seems less intuitive and does not provide any gain comparing with STO normalization." On the other hand, the authors of [6] found KST to perform slightly better than STO. Within our own group, we have also got inconclusive results with seemingly similar implementations of these algorithms, including finding that STO performs significantly better than KST.

This paper aims to clear up the confusion over the concepts of KST and STO, and reveal the cause of the difference in their performance. We recommend the use of "estimated KST" in the OpenKWS evaluation.

2. The Algorithms

2.1. Keyword Specific Thresholding (KST)

From the analysis of the evaluation metric, ATWV, we have seen that we need lower thresholds for rare keywords and higher thresholds for frequent keywords. To derive the threshold quantitatively, we make the following probabilistic assumption:

> **The KST Assumption:** *The raw score of a detection is the probability of it being correct.*

This is reasonable because the raw scores are calculated from the posterior probabilities of words in the lattices.

Suppose a detection has a raw score of p, and the corresponding keyword has a frequency of $N_{\text{true}}(w)$. Excluding this detection incurs a risk of miss of $\frac{1}{N} \cdot \frac{1}{N_{\text{true}}(w)} \cdot p$; retaining this detection incurs a risk of false alarm of $\frac{1}{N} \cdot \frac{\beta}{T - N_{\text{true}}(w)} \cdot (1-p)$. We should retain the detection if the latter risk is smaller, and exclude it vice versa. The optimal threshold should be the value of p that makes the two risks equal, which is given by the following formula:

$$\text{thr}(w) = \frac{\beta \cdot N_{\text{true}}(w)}{T + (\beta - 1) \cdot N_{\text{true}}(w)} \tag{3}$$

Fig. 2 shows that the optimal threshold increases monotonically with the keyword frequency $N_{\text{true}}(w)$, using the constants $\beta = 999.9$ and $T = 36{,}000$.

The values of $N_{\text{true}}(w)$ are actually unknown during spoken term detection, and estimated values must be used. We call Eq. (3) *oracle KST* if the oracle values of $N_{\text{true}}(w)$ are used, and *estimated KST* if estimated values are used instead.

Considering the KST assumption, a natural estimate of $N_{\text{true}}(w)$ will be the sum of the raw scores of all the detections of the keyword w. This sum has been called the *posterior sum*, and we'll denote it by $S(w)$. But how good is this estimate?

Fig. 3 shows the relationship between the estimate $S(w)$ and the true value $N_{\text{true}}(w)$, with each keyword represented as a dot. The values are calculated from a single system ("the Assamese system", which will be introduced in Section 3). The left panel shows the whole picture, and the right panel is a zoom-in of the part where both $S(w)$ and $N_{\text{true}}(w)$ are within 25. From the zoom-in we can see that, with the few outliers excluded, for most keywords we have $S(w) < N_{\text{true}}(w)$. This bias is due to the fact that many occurrences of keywords are not present in the index. To correct this bias, a boosting factor $\alpha > 1$ is multiplied to the posterior sum as the estimate for $N_{\text{true}}(w)$, i.e. $\hat{N}_{\text{true}}(w) = \alpha S(w)$. Substituting this into Eq. (3), we get the formula for estimated KST:

$$\text{thr}(w) = \frac{\beta \cdot \alpha \cdot S(w)}{T + (\beta - 1) \cdot \alpha \cdot S(w)} \tag{4}$$

The boosting factor α needs to be tuned on a validation corpus. As will be seen in Section 3, $\alpha = 1.5$ is a good choice.

2.2. Sum-to-One Score Normalization (STO)

Sum-to-One score normalization divides the raw score of each detection by the posterior sum $S(w)$ of the corresponding keyword to yield the normalized score. The name "sum-to-one" comes from the fact that for any keyword, the normalized scores of all the detections sum to one. It achieves the goal of boosting the detection score of rare keywords and suppressing the detection scores of frequent keywords because $S(w)$ is smaller for rare keywords and larger for frequent keywords.

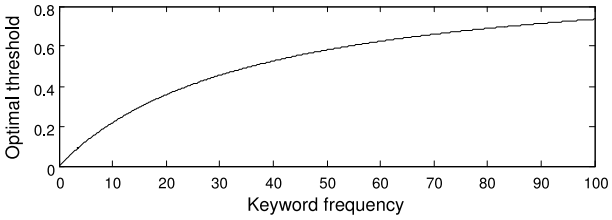

Figure 2: *The relationship between the optimal threshold and keyword frequency in KST*

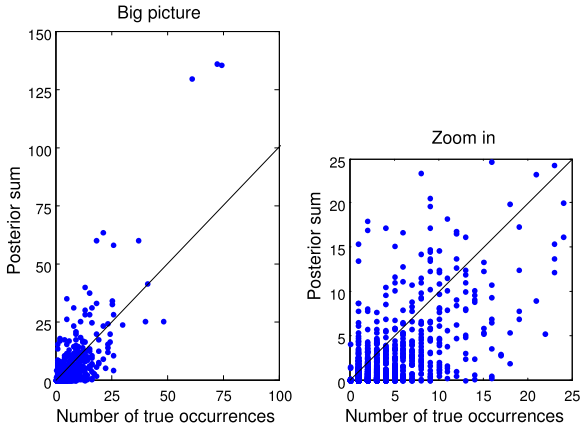

Figure 3: *The relationship between the posterior sum and the number of true occurrences*

The authors of [5] did not state a method to set the global threshold after score normalization. In practice, one can find the optimal global threshold empirically on a validation corpus. However, we can also borrow the assumption of KST and apply it to STO, by regarding the normalized scores as probabilities of the detections being correct. The posterior sum calculated from the normalized scores will then be $S(w) \equiv 1$ for all keywords, so Eq. (4) yields a global threshold for normalized scores:

$$\text{thr} = \frac{\beta \cdot \alpha}{T + (\beta - 1) \cdot \alpha} \tag{5}$$

2.3. The Connection Between KST and STO

Although KST and STO approach the need for keyword specific thresholds from different angles, their outcomes turn out similar. To better compare the two algorithms, we convert the global threshold for normalized scores given by STO in Eq. (5) back to keyword specific thresholds for raw scores, remembering that the raw score is the normalized score times $S(w)$:

$$\text{thr}(w) = \frac{\beta \cdot \alpha \cdot S(w)}{T + (\beta - 1) \cdot \alpha} \tag{6}$$

It is clear that the only difference between estimated KST (Eq. (4)) and STO (Eq. (6)) is the presence or absence of the factor $S(w)$ in the denominator. But this factor can make a big difference. As shown in Fig. 4 (with the constants $\beta = 999.9$, $T = 36{,}000$ and $\alpha = 1.5$), the estimated KST threshold is a hyperbolic function of $S(w)$, while the STO threshold is a linear function of $S(w)$. The latter can easily get larger than 1 when $S(w)$ is large, which causes the counter-intuitive phenomenon that frequent keywords are never detected. For rare keywords, estimated KST and STO exhibit similar behavior.

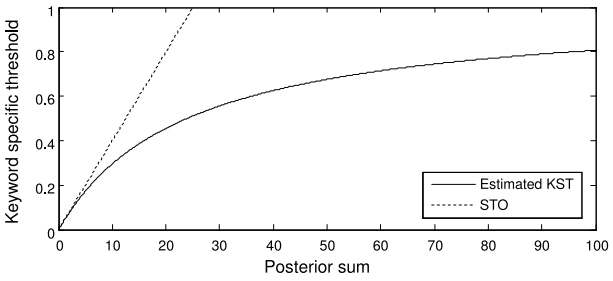

Figure 4: *Comparison of the estimated KST and STO keyword specific thresholds*

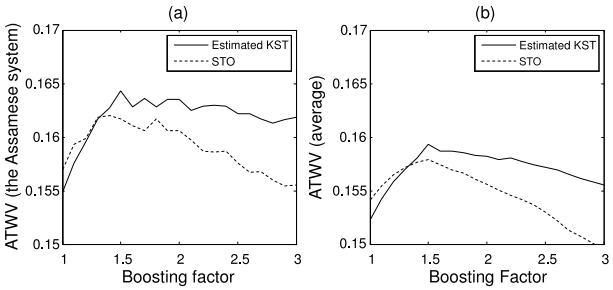

Figure 5: *Comparison of the estimated KST and STO keyword specific thresholds*

Algorithm	ATWV (single)	ATWV (average)
Oracle KST	0.1044	0.1066
Estimated KST	0.1643	0.1593
STO	0.1617	0.1579

Table 1: *Comparison of the ATWV of oracle KST, estimated KST and STO*

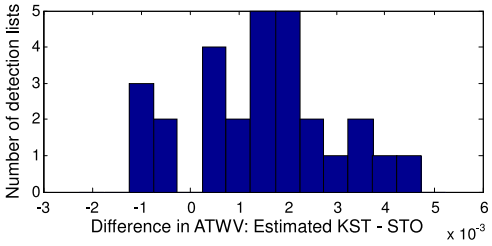

Figure 6: *Histogram of ATWV difference between estimated KST and STO on 28 systems*

3. Experiments and Analysis

3.1. Experimental Results

We compared the performance of oracle KST, estimated KST and STO using 28 spoken term detection systems. The systems came in 4 languages: Assamese, Bengali, Haitian Creole and Zulu[1]; for each language, we had 7 systems trained with different acoustic features (bottleneck features based on MFCC and log mel scale filterbank coefficients [7]), acoustic models (BNF-GMM trained with the bMMIE criterion and DNN-based acoustic model [8]), and speaker adaptation methods (feature-space constrained MLLR [9] and model-space MLLR [10]). All systems used a confusion network based index generated using the Janus toolkit with the Ibis decoder [11]. To study the global trend, we look at the average ATWV across all the 28 systems; to study the details, we examine a single Assamese system (which we refer to as "the Assamese system" hereafter). The ATWV numbers look low because these are single systems trained on only 10 hours of audio, which is the primary condition in this year's evaluation. When multiple systems are combined, they meet the performance goals of the IARPA Babel program, and provide state-of-the-art performance. We have also repeated the experiment on systems trained with 80 hours of audio, and our findings still hold true.

First, we investigated the effect of the boosting factor α on the ATWV in estimated KST and STO. Varying the boosting factor α from 1.0 to 3.0 with a step of 0.1, we plot the change of the ATWV of the Assamese system in Fig. 5(a). The curves appear very ragged, and it is unreliable to read off the optimal boosting factor from them. However, as shown in Fig. 5(b), the change of the average ATWV across all the 28 systems is much smoother. It happens that $\alpha = 1.5$ is the optimal value for both KST and STO, and with this boosting factor, estimated KST performs slightly better than STO.

Next, fixing the boosting factor α at 1.5, we compared the performance of all the three thresholding algorithms. Table 1 shows the ATWV of the Assamese system as well as the average ATWV across 28 systems. It is quite surprising that oracle KST, even though it has access to the real $N_{\text{true}}(w)$ values, performs a lot worse than estimated KST.

3.2. Significance Tests

The performance of estimated KST is only slightly better than that of STO. This gives rise to a natural question: Is the difference statistically significant?

[1]Babel OP1 language releases: `IARPA-babel{102b-v0.5a, 103b-v0.4b, 201b-v0.2b, 206b-v0.1e}`.

First, we compared the single-system performance of the two algorithms with two significance tests: the paired t-test, and the Wilcoxon signed-rank test [12]. In both tests, the "values" ($\frac{N_{\text{hit}}(w)}{N_{\text{true}}(w)} - \beta \cdot \frac{N_{\text{FA}}(w)}{T - N_{\text{true}}(w)}$) of each keyword yielded by the two systems are taken as a data pair. The paired t-test gave a p-value of 0.0183 (significant), but the Wilcoxon signed-ranked test gave a p-value of 0.7462 (insignificant). We choose to trust the result of the more rigorous Wilcoxon signed-rank test. This is because the paired t-test is based on the assumption that the differences of the two values in each data pair are normally distributed, which does not hold in reality.

Even though the difference between estimated KST and STO is not significant for a single system, we may compare the two algorithms on all the 28 systems. Now we regard the ATWV achieved by the two algorithms on each system as a data pair; a histogram of the ATWV differences on each system is shown in Fig. 6. As the differences are approximately normally distributed, it is safe to use the paired t-test. The test yielded a p-value of 1.66×10^{-5}. This means that estimated KST outperforms STO significantly at the system level (instead of the keyword level).

3.3. Why Estimated KST Is Better Than STO

Because the difference between estimated KST and STO on a single system is not significant, we study the cause of the difference by pooling the results of all the 28 systems. The difference between the average ATWV of estimated KST and

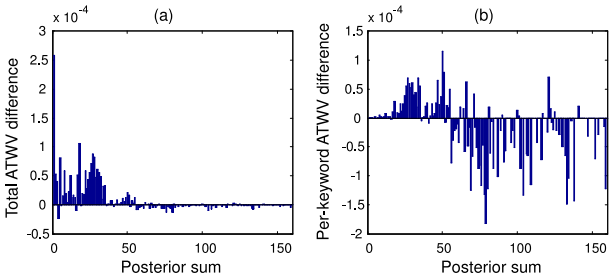

Figure 7: *Breakdown of the ATWV difference between estimated KST and STO (average over 28 systems) by posterior sum*

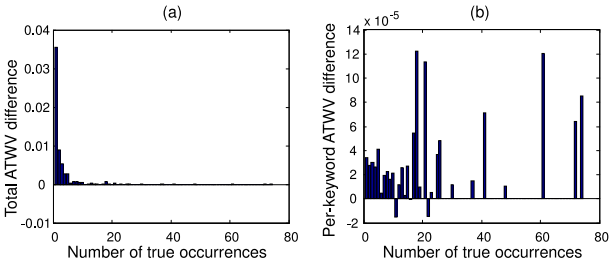

Figure 8: *Breakdown of the ATWV difference between estimated KST and oracle KST (on the Assamese system only) by number of true occurrences*

STO is 0.0014. We break this difference down by the posterior sum of the keywords, as shown in Fig. 7(a). The i-th bin is the total ATWV difference caused by keywords whose posterior sums fall with $[i-1, i)$. We observe a sharp positive peak in the first bin, and a wide positive peak in the range $20 \leq S(w) < 35$. When the posterior sum gets larger than 55, most bins are negative.

The second peak and the negative bins are meaningful; they can be understood with the help of Fig. 4, which shows the different keyword specific thresholds given by estimated KST and STO. Except for very rare keywords, STO gives higher thresholds than estimated KST, even larger than one. This has the positive effect of ruling out false alarms, but also has the negative effect of missing some true occurrences. Fig. 7(a) indicates that STO gains ATWV by avoiding false alarms for very frequent keywords ($S(w) \geq 55$), but loses ATWV from missing moderately frequent keywords ($20 \leq S(w) < 35$). In the OpenKWS evaluation, keywords are chosen in a way such that most keywords do not occur frequently (this applies to both the development keyword list, which we used in our experiment, and the evaluation keyword list), so STO loses more than it gains.

The peak in the first bin, despite having a high value, does not have as much area as the second peak. It occurs purely because many rare keywords fall in the bin $S(w) < 1$. If we divide the values of the bins by the number of keywords that fall in each bin, we can get the "per-keyword ATWV difference" for each bin (Fig. 7(b)). Now the peak in the first bin disappears, but the second peak and the negative bins remain.

3.4. Why Estimated KST Is Better Than Oracle KST

The difference between estimated KST and oracle KST is so large and significant that it can be easily explained with a single system. On the Assamese system, the ATWV difference between estimated KST and oracle KST is 0.0599. This time, we break down the difference by the number of true occurrences $N_{\text{true}}(w)$, as shown in Fig. 8(a). We see that more than half of the total difference is caused by keywords whose $N_{\text{true}}(w) = 1$. And this is not caused purely by a large number of keywords falling in this bin; even if we divide the values of the bins by the number of keywords falling in them (Fig. 8(b)), the first bin still has a significant positive value. This means that oracle KST loses to estimated KST mainly on the rare keywords.

The reason why oracle KST performs badly on rare keywords is the quantization of $N_{\text{true}}(w)$. Unlike $S(w)$, which can be arbitrarily small, $N_{\text{true}}(w)$ can only take on values of positive integers. According to Eq. (3), the minimum threshold possible is $\beta/(T + \beta - 1) \approx 0.027$. It turns out that this threshold is still often too high, because the majority of the detections have very small raw scores (ranging from 10^{-3} to 10^{-6}); the true detections are ruled out by the high threshold as well. Estimated KST beats oracle KST because $S(w)$ is a *continuous* estimate of $N_{\text{true}}(w)$. For rare keywords with low detection scores, $S(w)$ can be low commensurately. Even though this may produce some false alarms, for rare keywords, it is much more valuable to recall the true occurrences.

4. Conclusion

We have cleared up some confusion about keyword specific thresholding (KST) by differentiating between oracle KST and estimated KST. Oracle KST sets the thresholds based on the number of true occurrences $N_{\text{true}}(w)$ of keywords, which are actually not available; estimated KST sets the thresholds based on the posterior sum $S(w)$ of keywords and a boosting factor α. Estimated KST turns out to perform a lot better than oracle KST, because the former is able to set very low thresholds for rare keywords and recall their true occurrences.

We have also compared estimated KST with sum-to-one score normalization (STO). We have found that estimated KST performs slightly better than STO, which agrees with the discovery in [6]. The difference is insignificant on a single system, but significant when many systems are considered. The reason why estimated KST outperforms STO is that the thresholds set by STO for moderately frequent keywords are too high.

In practice, we recommend the use of estimated KST, because it has the best performance as well as a solid probabilistic foundation, and avoids the counter-intuitive phenomenon of never detecting any frequent keywords. For the OpenKWS evaluation, $\alpha = 1.5$ is a good value for the boosting factor, but one should tune it on a validation corpus to achieve the best performance.

5. Acknowledgements

This work was supported in part 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 purpose notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions annotated 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.

6. References

[1] L. Mangu, E. Brill and A. Stolcke, "Finding consensus among words: lattice-based word error minimization", in *Proc. of Eurospeech*, pp. 495-498, 1999.

[2] J. R. Rohlicek, W. Russell, S. Roukos and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting", in *Proc. of ICASSP*, pp. 627-630, 1989.

[3] NIST, "OpenKWS14 Evaluation Plan", 2014. Online: `http://www.nist.gov/itl/iad/mig/upload/KWS14-evalplan-v14.pdf`

[4] D. R. H. Miller, *et al.*, "Rapid and accurate spoken term detection", in *Proc. of InterSpeech*, pp. 314-317, 2007.

[5] J. Mamou, *et al.*, "System combination and score normalization for spoken term detection", in *Proc. of ICASSP*, pp. 8272-8276, 2013.

[6] D. Karakos, *et al.*, "Score normalization and system combination for improved keyword spotting", in *Proc. of ASRU*, pp. 210-215, 2013.

[7] J. Gehring, Y. Miao, F. Metze and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders", in *Proc. of ICASSP*, pp. 3377-3381, 2013.

[8] J. Gehring, Q. B. Nguyen, F. Metze and A. Waibel, "DNN acoustic modeling with modular multi-lingual feature extraction networks", in *Proc. of ASRU*, pp. 344-349, 2013.

[9] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition", Technical report, Cambridge University, 1997.

[10] C. J. Leggetter and P. C. Woodland, "Speaker adaptation of continuous density HMMs using multivariate linear regression", in Proc. of ICSLP, pp. 451-454, 1994.

[11] H. Soltau, F. Metze, C. Fügen and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment", in *Proc. of ASRU*, pp. 214-217, 2001.

[12] F. Wilcoxon, "Individual comparisons by ranking methods", in *Biometrics Bulletin*, vol. 1, no. 6, pp. 80-83, 1945.