MediaEval 2013 Spoken Web Search Task: System Performance Measures

L.J. Rodriguez-Fuentes, Mikel Penagarikano
Software Technologies Working Group (GTTS, http://gtts.ehu.es)
University of the Basque Country UPV/EHU
Department of Electricity and Electronics, ZTF/FCT
Barrio Sarriena, 48940 Leioa - SPAIN
luisjavier.rodriguez@ehu.es
mikel.penagarikano@ehu.es

May 30, 2013

1 Introduction

This document discusses how to measure system performance in the Spoken Web Search (SWS) task at MediaEval 2013. The discussion is based on different sources, including the NIST 2006 Spoken Term detection (STD) Evaluation Plan [1], the NIST 2010 Speaker Recognition Evaluation (SRE) Plan [2], the description of the scoring criteria applied in the SWS task at Mediaeval 2012 [3], the Albayzin 2012 Language Recognition Evaluation Plan [4] and the NIST 2013 Open Keyword Search (OpenKWS13) Evaluation Plan [5].

The SWS task at MediaEval 2013 is defined as *searching for audio content within audio content using an audio content query* [6]. The SWS task deals with two sets of multilingual speech contents: a set of query examples (involving one or more examples per query) and a set of audio documents on which searches are performed. Since both the queries and the audio documents may contain different languages, the search systems must be language-independent. Each query must be searched in an independent way, that is, without using information of other query searches. This also means that hard decisions must be taken separately for each query.

A perfect system would detect the exact locations of all the query occurrences in the audio documents, and would yield no false detections. As in SWS 2012 [7], the system output will consist of a list of query detections, including an audio document identifier, a query identifier, a starting time, a duration, a score indicating how likely the detection is (with more positive values indicating more likely occurrences) and a hard (`Yes/No`) decision. A reference file with the exact locations of all the queries within the audio documents will be used to measure system performance.

2 NIST 2006 STD performance measures

In the NIST 2006 STD Evaluation, detection accuracy was measured based on system decisions. Detection Error Tradeoff (DET) analysis was also carried out to report the maximum allowable detection accuracy and to evaluate the global system performance.

2.1 Counting hits and errors in system detections

Let us consider the list of query detections output by a given system. To determine which detections must be counted as hits and which ones must be counted as errors, the optimal alignment between the list of system detections and the list of actual query occurrences stored in the reference file is first computed (see [1], Section 4.5.1).

The procedure can be summarized as follows: (1) a system detection of a query is aligned with an actual occurrence of that query if its mid point is less than or equal to 0.5 seconds (tolerance interval) from the time span of that occurrence; (2) a one-to-one mapping is defined: if two system detections can be aligned with the same actual occurrence, only one of them will be aligned with it; similarly, if two actual occurrences of a given query can be aligned with the same system detection, only one of them will be alignd with it; and (3) within these constraints, the alignment is performed so as to maximize the number of alignments between system detections and actual occurrences.

The alignment file not only includes all system detections, but also the actual query occurrences not detected by the system. Starting from the alignments, system decisions can be evaluated in terms of two types of errors: (1) *misses* (actual query occurrences that have not been detected by the system); and (2) *false alarms* (system detections that do not match any actual query occurrence). Errors are counted according to the following criteria:

- A system detection aligned with an actual occurrence is counted as a *hit* if the system decision is `Yes`, and as a miss error if the system decision is `No`.
- A system detection not aligned with any actual occurrence is counted as a false alarm error if the system decision is `Yes`, and not counted at all if the system decision is `No`.
- Finally, all the actual query occurrences not aligned with any system detection are counted as miss errors.

System decisions are assumed to be made by applying a threshold θ to scores: the decision is `Yes` if the score is greater than or equal to θ, and `No` otherwise. This means that hard decisions included in the output file should correspond to a particular threshold θ_{act}. DET analysis evaluates system performance for a wide range of thresholds, yielding as a byproduct the optimal threshold θ_{opt}. Hereafter, we will consider the decisions corresponding to a generic threshold θ.

2.2 Miss and false alarm error rates

The miss error rate for a query q and a threshold θ is computed as:

$$P_{\text{miss}}(q,\theta) = \frac{N_{\text{miss}}(q,\theta)}{N_{\text{act}}(q)} \tag{1}$$

where $N_{\text{miss}}(q,\theta)$ is the number of miss errors corresponding to query q and threshold θ, and $N_{\text{act}}(q)$ is the number of actual occurrences of query q (i.e. the number of *target trials*) in the audio documents[1].

The false alarm error rate cannot be computed so easily, because the set of *non-occurrences* of each query q (i.e. the set of *non-target trials*) is not explicitly defined. However, a new parameter n_{tps}: number of trials per second, can be arbitrarily defined (typically, $n_{\text{tps}} = 1$), so that the number of trials for any single query is $N = n_{\text{tps}} \cdot T_{\text{audio}}$, where T_{audio} is the total duration of the audio documents used for testing. Then, the number of non-target trials (i.e. the maximum number of false alarms) is:

$$N_{\text{nt}}(q) = N - N_{\text{act}}(q) = n_{\text{tps}} \cdot T_{\text{audio}} - N_{\text{act}}(q) \tag{2}$$

Finally, the false alarm error rate for query q and threshold θ is computed as:

$$P_{\text{fa}}(q,\theta) = \frac{N_{\text{fa}}(q,\theta)}{N_{\text{nt}}(q)} \tag{3}$$

where $N_{\text{fa}}(q,\theta)$ is the number of false alarm errors corresponding to query q and threshold θ. Note that both $P_{\text{miss}}(q,\theta)$ and $P_{\text{fa}}(q,\theta)$ are comprised between 0 and 1.

Then, the average error rates over the whole set of queries $\mathcal{Q}$ (assuming that all the queries are equally likely) can be computed as follows:

$$P_{\text{miss}}(\theta) = \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} P_{\text{miss}}(q,\theta) \tag{4}$$

$$P_{\text{fa}}(\theta) = \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} P_{\text{fa}}(q,\theta) \tag{5}$$

2.2.1 On the definition of non-target trials

In the above described procedure, virtual non-target trials have been defined by means of the parameter n_{tps}, which arbitrarily determines the time span of a trial (e.g. one second). This approach was originally defined for the NIST 2006 STD evaluation and followed in Mediaeval 2012 SWS and NIST 2013 OpenKWS evaluations.

An attempt to overcome the definition of virtual non-target trials has been made in NIST 2013 OpenKWS Evaluation, where the segments obtained from speech/non-speech detection are alternatively used to define the set of trials [5]. For any given query, a target trial is defined as a speech segment that contains one or more actual occurrences of that query, the remaining speech and non-speech segments being non-target trials. A similar approach was already applied

[1] We assume that $N_{\text{act}}(q) > 0 \ \forall q$.

in Mediaeval 2011 SWS, where a target trial was defined as an audio document containing one or more occurrences of the considered query, the remaining documents being non-target trials. However, under this approach the exact position of the query inside the audio segment is not taken into account, and thus the ability of systems to match the exact position of a query occurrence inside an audio document is not evaluated. Since this ability is key for the SWS task, we keep the parameter n_{tps} and use virtual non-target trials to compute the false alarm error rate.

2.3 Term-Weighted Value

The so called *Term-Weighted Value* (TWV) is defined as a weighted combination of the miss and false alarm error rates, averaged over the set of queries, as follows:

$$\begin{aligned} \text{TWV}(\theta) &= 1 - \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} \left(P_{\text{miss}}(q,\theta) + \beta \cdot P_{\text{fa}}(q,\theta) \right) \\ &= 1 - \left(P_{\text{miss}}(\theta) + \beta \cdot P_{\text{fa}}(\theta) \right) \end{aligned} \tag{6}$$

The weight factor $\beta > 0$ is defined as:

$$\beta = \frac{C_{\text{fa}} \cdot (1 - P_{\text{target}})}{C_{\text{miss}} \cdot P_{\text{target}}} \tag{7}$$

where $C_{\text{miss}} > 0$ and $C_{\text{fa}} > 0$ are the costs of miss and false alarm errors, respectively, and $P_{\text{target}} \in [0, 1]$ is the prior probability of a target trial (which is assumed to be constant across queries). $\text{TWV}(\theta)$ ranges from $-\beta$ to 1, being 1 for a perfect system, 0 for a simple system making always the decision `No` (i.e. rejecting all the trials) and $-\beta$ for the worst possible system.

In the NIST 2006 STD Evaluation, the TWV for system hard decisions $\text{TWV}(\theta_{\text{act}})$, known as *Actual Term-Weighted Value* (ATWV), was used as primary evaluation measure. The *Maximum Term-Weighted Value* (MTWV), defined as $\text{TWV}(\theta_{\text{opt}})$, was also reported after DET analysis of system scores. Note that MTWV defines an upper bound for ATWV: if the system was perfectly calibrated, ATWV should equal MTWV.

2.4 Derivation of TWV from the NIST SRE normalized cost function

The terminology employed to define the TWV in this report differs from that used in [1]: instead of C, C_{fa} is used; instead of V, C_{miss} is used[2]; and instead of Pr_{term}, P_{target} is used. The reason for these changes is that TWV can be better understood by showing its relation to the cost function defined in the NIST 2010 Speaker Recognition Evaluation [2].

In NIST SRE, the test set consists of audio segments, each containing speech from a single speaker. These segments play the same role as either the actual query occurrences or the virtual query non-occurrences considered in the SWS task. Similarly, the set of target speakers play the same role as queries. Finally,

[2] In fact, the use of V (*value* of correct detections) in [1] is not well-motivated, since it is not the rate of correct detections but the rate of misses what it deals with.

a trial is defined in NIST SRE by the question "Is target speaker `s` speaking in the test audio segment `t`?", which is equivalent to "Does this detection of query `q` match an actual occurrence of query `q`?". In both cases, speaker/query occurrences must be detected independently from other speakers/queries. From this point of view, SWS and NIST SRE tasks can be evaluated in the same way.

Let us consider the Detection Cost function C_{Det} as defined in NIST 2010 SRE [2]:

$$C_{\text{Det}} = C_{\text{miss}} \cdot P_{\text{miss}} \cdot P_{\text{target}} + C_{\text{fa}} \cdot P_{\text{fa}} \cdot (1 - P_{\text{target}}) \tag{8}$$

where P_{miss} and P_{fa} are the miss and false alarm error rates given by system hard decisions (this definition can be easily generalized to the case of decisions based on a threshold θ). A normalized measure was defined in NIST 2010 SRE by dividing C_{Det} by the best cost that could be obtained with a trivial system that gives always the same response (i.e. always rejecting or always accepting the trials, whichever gives the lower cost):

$$C_{\text{Default}} = \min \begin{cases} C_{\text{miss}} \cdot P_{\text{target}} \\ \\ C_{\text{fa}} \cdot (1 - P_{\text{target}}) \end{cases} \tag{9}$$

$$C_{\text{Norm}} = \frac{C_{\text{Det}}}{C_{\text{Default}}} \tag{10}$$

In most practical cases, where the target prior is low and costs are not highly biased, $C_{\text{Default}} = C_{\text{miss}} \cdot P_{\text{target}}$ (corresponding to a trivial system that rejects all the trials), and the *Normalized Detection Cost function* will be:

$$C_{\text{Norm}} = P_{\text{miss}} + \frac{C_{\text{fa}} \cdot (1 - P_{\text{target}})}{C_{\text{miss}} \cdot P_{\text{target}}} \cdot P_{\text{fa}} \tag{11}$$

Clearly, the Term-Weighted Value is given by $\text{TWV} = 1 - C_{\text{Norm}}$. By the way, this provides a suitable explanation for β and the terminology used in this report.

2.5 Application-dependent parameters and DET analysis

The parameters C_{miss}, C_{fa} and P_{target} determine an application of interest (an operating point) for which the system should be optimized. In NIST 2006 STD, $C_{\text{miss}} = 10$, $C_{\text{fa}} = 1$ and $P_{\text{target}} = 10^{-4}$, thus $\beta = 999.9$. Why this particular operating point? Given two systems A and B, do performance differences at that point express the overall difference in performance between A and B? Note that a system A may yield better TWV performance than a system B at a given operating point, but their roles may change at another operating point. As we noted above, to allow a global performance comparison of two systems, DET curves are built with the set of pairs $(P_{\text{miss}}(\theta), P_{\text{fa}}(\theta))$ obtained for a wide range of thresholds.

2.6 A simple interpretation of C_{Norm}

In this section, we will show that C_{Norm} has a very simple interpretation in terms of miss and false alarm errors and costs, provided that P_{target} is the empirical prior of target trials.

Let N be the number of trials considered for any single query, $N_{\text{true}}(q)$ the number of target trials for a query q and N_{true} the total number of target trials. Let $N_{\text{miss}}(q)$ and $N_{\text{fa}}(q)$ be the number of miss and false alarm errors for a query q, respectively. And let N_{miss} and N_{fa} be the total number of miss and false alarm errors, respectively. We will assume that target trials are uniformly distributed in $\mathcal{Q}$, that is, $\forall q \in \mathcal{Q}$, it holds $N_{\text{true}}(q) = N_{\text{true}}/|\mathcal{Q}|$. Then, the empirical prior of target trials for any query q is given by:

$$P_{\text{target}}^{(\text{emp})}(q) = \frac{N_{\text{true}}(q)}{N} = \frac{N_{\text{true}}}{|\mathcal{Q}| \cdot N} = P_{\text{target}}^{(\text{emp})} \tag{12}$$

In the following, without loss of generality, the empirical prior of any query q will be called $P_{\text{target}}^{(\text{emp})}$. Then, P_{miss} and P_{fa} can be expressed as:

$$\begin{aligned} P_{\text{miss}} &= \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} P_{\text{miss}}(q) = \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} \frac{N_{\text{miss}}(q)}{N_{\text{true}}(q)} = \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} \frac{N_{\text{miss}}(q)}{N \cdot P_{\text{target}}^{(\text{emp})}} \\ &= \frac{N_{\text{miss}}}{|\mathcal{Q}| \cdot N \cdot P_{\text{target}}^{(\text{emp})}} \end{aligned} \tag{13}$$

$$\begin{aligned} P_{\text{fa}} &= \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} P_{\text{fa}}(q) = \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} \frac{N_{\text{fa}}(q)}{N - N_{\text{true}}(q)} = \frac{1}{|\mathcal{Q}|} \sum_{\forall q \in \mathcal{Q}} \frac{N_{\text{fa}}(q)}{N - N \cdot P_{\text{target}}^{(\text{emp})}} \\ &= \frac{N_{\text{fa}}}{|\mathcal{Q}| \cdot N \cdot (1 - P_{\text{target}}^{(\text{emp})})} \end{aligned} \tag{14}$$

Introducing Eqs. 13 and 14 in Eq. 11, the normalized detection cost function can be expressed as:

$$\begin{aligned} C_{\text{Norm}} &= \frac{1}{|\mathcal{Q}| \cdot N \cdot P_{\text{target}}^{(\text{emp})}} \left(N_{\text{miss}} + \frac{P_{\text{target}}^{(\text{emp})}}{P_{\text{target}}} \cdot \frac{1 - P_{\text{target}}}{1 - P_{\text{target}}^{(\text{emp})}} \cdot \frac{C_{\text{fa}}}{C_{\text{miss}}} \cdot N_{\text{fa}} \right) \\ &= \frac{1}{N_{\text{true}}} \left(N_{\text{miss}} + \frac{P_{\text{target}}^{(\text{emp})}}{P_{\text{target}}} \cdot \frac{1 - P_{\text{target}}}{1 - P_{\text{target}}^{(\text{emp})}} \cdot \frac{C_{\text{fa}}}{C_{\text{miss}}} \cdot N_{\text{fa}} \right) \end{aligned} \tag{15}$$

Finally, if $P_{\text{target}} = P_{\text{target}}^{(\text{emp})}$, we get:

$$C_{\text{Norm}} = \frac{1}{N_{\text{true}}} \left(N_{\text{miss}} + \frac{C_{\text{fa}}}{C_{\text{miss}}} \cdot N_{\text{fa}} \right) \tag{16}$$

By inspecting Eq. 16, we find that the ratio $C_{\text{fa}}/C_{\text{miss}}$ controls the relative weight that is given to false alarm errors with regard to miss errors in the cost function. In particular, if $C_{\text{fa}} = C_{\text{miss}}$, miss and false alarm errors have the same weight in the computation of C_{Norm} (and thus, of TWV). Note also that a trivial system rejecting all the trials yields $N_{\text{fa}} = 0$ and $N_{\text{miss}} = N_{\text{true}}$, so that $C_{\text{Norm}} = 1$ (and TWV = 0).

3 Mediaeval 2012 SWS performance measure

For the SWS task at Mediaeval 2012, a modified TWV was proposed as performance measure [3], aiming to address usage scenarios (i.e. applications) where the false alarms were less penalised than in the TWV as defined for the NIST 2006 STD Evaluation. The operating point of the TWV function was changed so that systems tended to output more positive detections, leading to higher recall figures (i.e. more actual query ocurrences among the retrieved items). The new operating point was given by $C_{\text{miss}} = C_{\text{fa}} = 1$ and:

$$\begin{aligned} P_{\text{target}} &= |\mathcal{Q}| \cdot P_{\text{target}}^{(\text{emp})} \\ &= |\mathcal{Q}| \cdot \frac{N_{\text{true}}}{|\mathcal{Q}| \cdot N} = \frac{N_{\text{true}}}{n_{\text{tps}} \cdot T_{\text{audio}}} \end{aligned} \tag{17}$$

where $P_{\text{target}}^{(\text{emp})}$ represents the empirical prior of target trials, N_{true} is the number of target trials $\forall q \in \mathcal{Q}$, and $N = n_{\text{tps}} \cdot T_{\text{audio}}$ is the number of trials for any single query $q \in \mathcal{Q}$. According to the above defined prior and costs, the factor β weighting P_{fa} in the modified TWV is given by:

$$\beta = \frac{n_{\text{tps}} \cdot T_{\text{audio}} - N_{\text{true}}}{N_{\text{true}}} \tag{18}$$

The main issue with the modified TWV is that P_{target} (and thus, β) depends on the set of queries $\mathcal{Q}$ and the set of audio documents Ω used for testing, since $N_{\text{true}} = N_{\text{true}}(\mathcal{Q}, \Omega)$ and $T_{\text{audio}} = T_{\text{audio}}(\Omega)$. This means that a different P_{target} must be computed for each pair $(\mathcal{Q}, \Omega)$ and a different operating point is therefore used to compute system performance in each case.

Now, the analysis carried out in Section 2.6 is resumed in order to provide a suitable interpretation of the modified TWV. Assuming that $C_{\text{miss}} = C_{\text{fa}} = 1$ and introducing Eq. 17 in Eq. 15, we get:

$$\begin{aligned} C_{\text{Norm}} &= \frac{1}{N_{\text{true}}} \left(N_{\text{miss}} + \frac{1}{|\mathcal{Q}|} \cdot \frac{1 - |\mathcal{Q}| \cdot P_{\text{target}}^{(\text{emp})}}{1 - P_{\text{target}}^{(\text{emp})}} \cdot N_{\text{fa}} \right) \\ &= \frac{1}{N_{\text{true}}} \left(N_{\text{miss}} + \frac{1}{|\mathcal{Q}| \cdot \alpha} \cdot N_{\text{fa}} \right) \end{aligned} \tag{19}$$

where[3]:

$$\alpha = \frac{1 - P_{\text{target}}^{(\text{emp})}}{1 - |\mathcal{Q}| \cdot P_{\text{target}}^{(\text{emp})}} \tag{20}$$

We conclude that using $C_{\text{miss}} = C_{\text{fa}} = 1$ and $P_{\text{target}} = |\mathcal{Q}| \cdot P_{\text{target}}^{(\text{emp})}$, as it was done in SWS 2012, implies that miss errors have $|\mathcal{Q}| \cdot \alpha$ times as much weight as false alarm errors in the computation of C_{Norm} (and thus, of TWV). Besides, the factor $|\mathcal{Q}| \cdot \alpha$ depends on both the set of queries $\mathcal{Q}$ and the dataset Ω used for testing. In other words, the relative weight of miss and false alarm errors in TWV computation is different for each pair $(\mathcal{Q}, \Omega)$.

[3] Note that $\alpha > 1$, since we assume that $P_{\text{target}} = |\mathcal{Q}| \cdot P_{\text{target}}^{(\text{emp})} < 1$.

4 Mediaeval 2013 SWS performance measures

In this Section, we propose two (primary and alternative) performance measures and a secondary measure characterizing the required amount of processing resources, to be used in Mediaeval 2013 SWS.

4.1 Evaluating system decisions: ATWV and MTWV

For compatibility with previous evaluations, the primary performance measure in Mediaeval 2013 SWS should be based on system hard decisions. The *Actual Term-Weighted Value* (ATWV) seems the best choice, for two reasons:

1. ATWV not only takes into account miss and false alarm error rates but also the prior of target trials and the miss and false alarm error costs, that allow focusing on a particular application.

2. ATWV has been used as primary measure in the NIST 2006 STD, NIST 2013 OpenKWS and Mediaeval 2011 and 2012 SWS evaluations.

Besides the ATWV, the *Maximum Term-Weighted Value* (MTWV), i.e. the maximum TWV that can be attained based on system scores, along with the corresponding DET curve, could be computed after DET analysis, to get an estimation of the system performance and to detect calibration issues at the given operating point. Note that, in order to allow meaningful DET analyses, participants should be encouraged to produce detections far beyond the threshold applied for making system hard decisions.

Regarding the application parameters, we suggest setting the prior of target trials to a value that approximately reflects the empirical prior that users may expect to find in the set of audio documents, and the ratio $C_{\text{fa}}/C_{\text{miss}}$ to a value that reflects the desired weight of both types of errors. For Mediaeval 2013 SWS, we suggest:

$$\left.\begin{array}{rcl} P_{\text{target}} & = & 0.00015 \\ C_{\text{fa}} & = & 1 \\ C_{\text{miss}} & = & 100 \end{array}\right\} \tag{21}$$

which means that misses will have 100 times as much weight as false alarms in the computation of TWV (see Eq. 16). According to these values, $\beta = 66.66$ (it was not our intention to set such a satanic β).

4.2 Evaluating system scores: C_{nxe} and $C_{\text{nxe}}^{\text{min}}$

4.2.1 The effective prior as a single application parameter

The C_{Norm} of Eq. 11 depends on the error rates P_{miss} and P_{fa}, and the application dependent parameters C_{miss}, C_{fa} and P_{target}. The three application dependent parameters can be represented by a single parameter, $P_{\text{tar}} = P_{\text{tar}}(C_{\text{miss}}, C_{\text{fa}}, P_{\text{target}})$, called *effective prior*, defined as follows:

$$\frac{C_{\text{fa}} \cdot (1 - P_{\text{target}})}{C_{\text{miss}} \cdot P_{\text{target}}} = \frac{(1 - P_{\text{tar}})}{P_{\text{tar}}} \tag{22}$$

The effective prior is thus given by:

$$\begin{aligned} P_{\text{tar}} &= \frac{C_{\text{miss}} \cdot P_{\text{target}}}{C_{\text{miss}} \cdot P_{\text{target}} + C_{\text{fa}} \cdot (1 - P_{\text{target}})} \\ &= \frac{1}{1 + \frac{C_{\text{fa}} \cdot (1 - P_{\text{target}})}{C_{\text{miss}} \cdot P_{\text{target}}}} \end{aligned} \tag{23}$$

and the C_{Norm} can be rewritten in terms of the effective prior as:

$$C_{\text{Norm}} = P_{\text{miss}} + \frac{(1 - P_{\text{tar}})}{P_{\text{tar}}} \cdot P_{\text{fa}} \tag{24}$$

That is, any operating point $(C_{\text{miss}}, C_{\text{fa}}, P_{\text{target}})$ has an equivalent point $(1, 1, P_{\text{tar}})$ with a shifted target prior and flat error costs. In what follows, the single parameter P_{tar} will be used instead of the triplet $(C_{\text{miss}}, C_{\text{fa}}, P_{\text{target}})$. In the case of the operating point suggested for the 2013 SWS evaluation (see Eq. 21), $P_{\text{tar}} = 0.0148$.

4.2.2 On the goodness of well calibrated log-likelihood-ratio scores

The primary performance measure for Mediaeval 2013 SWS, the TWV, is based on hard decisions. Given a set of system scores, a threshold is applied on them to make system decisions. Under this framework, the evaluee must guess the optimal threshold, i.e. the threshold yielding the MTWV.

From a probabilistic point of view, it is possible to estimate the expected cost of a decision $d \in \{\text{true}, \text{false}\}$ for a single trial $t = (x, q)$ consisting of a segment x and a query q:

$$\begin{aligned} E_{d=\text{true}}[\text{Cost}] &= P(H_1|x) \\ E_{d=\text{false}}[\text{Cost}] &= P(H_0|x) \end{aligned} \tag{25}$$

where $P(H_0|x)$ is the probability of the *null* hypothesis (the segment x contains the query q) and $P(H_1|x)$ is the probability of the *alternative* hypothesis (the segment x does not contain the query q). The decision of the system should be the one with the minimum expected cost, i.e.:

$$d = \text{true} \quad \Leftrightarrow \quad P(H_1|x) < P(H_0|x) \tag{26}$$

The posterior ratio is defined as:

$$\frac{P(H_0|x)}{P(H_1|x)} = \frac{P(x|H_0) \cdot P(H_0)}{P(x|H_1) \cdot P(H_1)} = lr_t \cdot \frac{P_{\text{tar}}}{(1 - P_{\text{tar}})} \tag{27}$$

where:

$$lr_t = \frac{P(x|H_0)}{P(x|H_1)} \tag{28}$$

is the likelihood ratio corresponding to trial t. Whe can now rewrite Eq. 26 as:

$$d = \text{true} \quad \Leftrightarrow \quad lr_t > \frac{(1 - P_{\text{tar}})}{P_{\text{tar}}} \tag{29}$$

That is, minimum expected cost decisions can be made based only on the application independent likelihood ratio $lr_t \in [0, \infty]$ and the application dependent effective prior P_{tar}. Eq. 29 is often written in logarithmic form:

$$d = \text{true} \quad \Leftrightarrow \quad llr_t > -\text{logit}(P_{\text{tar}}) \tag{30}$$

where $llr_t = \log(lr_t)$ is the log-likelihood ratio corresponding to trial t and $\text{logit}(p) = \log \frac{p}{(1-p)}$ is the log odds ratio function.

If the output scores of a system were *well calibrated* log-likelihood ratios ($llr_t \in [-\infty, \infty]$), then the minimum expected cost threshold, θ_{mincost}, would depend only on application parameters:

$$\theta_{\text{mincost}} = -\text{logit}(P_{\text{tar}}) \tag{31}$$

Traditionally, the threshold θ is estimated by minimizing the cost function (or maximizing the value function) on a development set. Therefore, building a system for a different operating point implies doing a new optimization of the threshold[4].

In contrast, a system that outputs log-likelihood-ratio scores could be adapted to almost any operating point just by applying the minimum expected cost threshold, θ_{mincost}. The latter does not mean that the development set is no longer needed. In fact, it is used to calibrate the system output (ensuring proper log-likelihood ratios). Once a system has been calibrated, it can be used to make decisions at any operating point without further optimization.

4.2.3 The normalized cross entropy: C_{nxe}

As pointed out before, if the scores of a SWS system represent log-likelihood ratios, they may be viewed as more informative and useful for a range of possible applications. Moreover, it is possible to delve into how much information they provide about the ground truth of an evaluation set.

Let us assume that the system under evaluation, $\mathcal{S}$, submits a set of log-likelihood ratios llr_t for a set of trials $T(\mathcal{S})$, each trial $t = (x, q) \in T(\mathcal{S})$ consisting of a segment x and a query q. Given the ground truth of a trial t, $\mathcal{G}_t \in \{\text{true}, \text{false}\}$, the goodness of llr_t can be measured by the logarithmic cost function [4]:

$$C_{\log}(llr_t) = -\log P(\mathcal{G}_t|llr_t) \tag{32}$$

That is, when the system favours the ground truth, then $P(\mathcal{G}_t|llr_t) \approx 1$, so that $C_{\log}(llr_t) \approx 0$, but if it favours the opposite, then $P(\mathcal{G}_t|llr_t) \approx 0$ and $C_{\log}(llr_t) \gg 0$.

The posterior of the ground truth can be written in terms of the probabilities of the *null* and *alternative* hypotheses of Eq. 25, as follows:

$$P(\mathcal{G}_t|llr_t) = \begin{cases} P(H_0|x) & t \in T_{\text{true}}(\mathcal{S}) \\ \\ P(H_1|x) & t \in T_{\text{false}}(\mathcal{S}) \end{cases} \tag{33}$$

where $T_{\text{true}}(\mathcal{S}) = \{t \in T(\mathcal{S}) | \mathcal{G}_t = \text{True}\}$ is the set of target trials and $T_{\text{false}}(\mathcal{S}) = \{t \in T(\mathcal{S}) | \mathcal{G}_t = \text{false}\}$ is the set of non-target trials. From Eq. 27 and taking into account that $P(H_1|x) = (1 - P(H_0|x))$, it follows:

$$\frac{P(H_0|x)}{1 - P(H_0|x)} = lr_t \cdot \frac{P_{\text{tar}}}{(1 - P_{\text{tar}})} = e^{llr_t + \text{logit}(P_{\text{tar}})} \tag{34}$$

[4] As pointed out in Section 3, in Mediaeval 2012 SWS, the operating points of the development and evaluation sets were different, so the threshold optimized on the development set, θ_{devel}, may not match the *a priori* optimal threshold on the evaluation set, θ_{eval}.

$$P(H_0|x) = \frac{1}{1+e^{-(llr_t+\text{logit}(P_{\text{tar}}))}} = \text{sigmoid}(llr_t + \text{logit}(P_{\text{tar}})) \tag{35}$$

In a similar manner, for the *alternative* hypothesis, it holds:

$$P(H_1|x) = \frac{1}{1+e^{llr_t+\text{logit}(P_{\text{tar}})}} = \text{sigmoid}(-(llr_t + \text{logit}(P_{\text{tar}}))) \tag{36}$$

Then, the logarithmic cost function will be:

$$C_{\log}(llr_t) = \begin{cases} -\log(\text{sigmoid}(llr_t + \text{logit}(P_{\text{tar}}))) & t \in T_{\text{true}}(\mathcal{S}) \\ -\log(\text{sigmoid}(-(llr_t + \text{logit}(P_{\text{tar}})))) & t \in T_{\text{false}}(\mathcal{S}) \end{cases} \tag{37}$$

Note that, as the logarithmic cost function is based on the posterior probability, it depends both on the log-likelihood ratio llr_t and the prior probability P_{tar}. In previous NIST SRE evaluations [2], a non-informative flat prior $P_{\text{tar}} = 0.5$ was used intending to set an application independent cost function[5]. Nevertheless, with regard to SWS, it is not clear which should be the application independent non-informative prior, since queries are drawn from a vocabulary of size $|V|$ (assuming single-word queries) and flat error costs are used ($C_{\text{miss}} = C_{\text{fa}} = 1$). Under these conditions, the effective prior would be $P_{\text{tar}} = \frac{1}{|V|} \ll 0.5$.

If we average over the full set of trials and divide by $\log 2$, we get the so called *empirical cross entropy* (in information bits):

$$C_{\text{xe}} = \frac{1}{\log 2} \cdot \left(\frac{P_{\text{tar}}}{|T_{\text{true}}(\mathcal{S})|} \sum_{t \in T_{\text{true}}(\mathcal{S})} C_{\log}(llr_t) + \frac{1-P_{\text{tar}}}{|T_{\text{false}}(\mathcal{S})|} \sum_{t \in T_{\text{false}}(\mathcal{S})} C_{\log}(llr_t) \right) \tag{38}$$

The empirical cross entropy of a system can be normalized by comparing it to that of a trivial system that gives always non-informative scores (i.e. $llr_t = 0 \ \ \forall t$). The empirical cross entropy of such a trivial system, sometimes called the *prior entropy*, is given by:

$$C_{\text{xe}}^{prior} = \frac{1}{\log 2} \cdot \left(P_{\text{tar}} \cdot \log \frac{1}{P_{\text{tar}}} + (1-P_{\text{tar}}) \cdot \log \frac{1}{1-P_{\text{tar}}} \right) \tag{39}$$

Finally, the normalized empirical cross entropy is defined as:

$$C_{\text{nxe}} = \frac{C_{\text{xe}}}{C_{\text{xe}}^{prior}} \tag{40}$$

The normalized cross entropy measures the apparent knowledge that the SWS system has on the ground truth. Specifically, it accounts for the fraction of information that is not provided by system scores. A perfect system would get $C_{\text{nxe}} \approx 0$ and a non-informative system would get $C_{\text{nxe}} = 1$, whereas $C_{\text{nxe}} > 1$ would indicate a severe misscalibration of the log-likelihood ratio scores.

[5] Note that if $P_{\text{tar}} = 0.5$, then $C_{\log}(llr_t) = \begin{cases} -\log(\frac{1}{1+\frac{1}{lr_t}}) & t \in T_{\text{true}}(\mathcal{S}) \\ -\log(\frac{1}{1+lr_t}) & t \in T_{\text{false}}(\mathcal{S}) \end{cases}$

4.2.4 Evaluating censored subsets of system scores

Note that the set of trials $T(\mathcal{S})$ depends on the system $\mathcal{S}$ itself. Each evaluee is free to decide when to stop the process of finding query matches. The 2013 SWS evaluation metric assumes that there can be up to n_{tps} trials per second for each query, which implies that the size of the entire set of trials T will be:

$$|T| = |\mathcal{Q}| \cdot n_{\text{tps}} \cdot T_{\text{audio}} \tag{41}$$

However, the trials submitted by a system $\mathcal{S}$ will be a subset of the entire set of trials: $T(\mathcal{S}) \subset T$, and it will typically be $|T(\mathcal{S})| \ll |T|$. In other words, the evaluee submits a *censored* subset of scores.

In order to compare the performance of two systems, they must refer to the same ground truth, i.e. the same set of trials. Therefore, the evaluator must do a reasonable guess of the censored scores. It seems fair to assume that those missing scores are lower than the minimum submited score[6]. Then, an optimistic guess would be[7]:

$$\begin{aligned} \forall t \notin T(\mathcal{S}), \quad llr_t &= llr_{\min}(\mathcal{S}) \\ &= \min\{llr_{t'} | t' \in T(\mathcal{S})\} \end{aligned} \tag{42}$$

Now, the empirical cross entropy can be computed for the entire set of trials T as follows:

$$C_{\text{xe}} = \frac{1}{\log 2} \cdot \left(\frac{P_{\text{tar}}}{|T_{\text{true}}|} \sum_{t \in T_{\text{true}}} C_{\log}(llr_t) + \frac{1 - P_{\text{tar}}}{|T_{\text{false}}|} \sum_{t \in T_{\text{false}}} C_{\log}(llr_t) \right) \tag{43}$$

where T_{true} and T_{false} represent the entire sets of target and non-target trials, respectively.

4.2.5 Evaluating the miscalibration: $C_{\text{nxe}}^{\min}$

The cross entropy measures both discrimination (between target and non-target trials) and calibration. To estimate the calibration loss, the evaluator can optimally recalibrate a system using a simple reversible transformation, such as [8]:

$$\hat{llr}_t = \gamma \cdot llr_t + \delta \tag{44}$$

where γ and δ are calibration parameters that can be used to minimize the normalized cross entropy:

$$C_{\text{nxe}}^{\min} = \min_{\gamma,\delta} \left\{ \hat{C}_{\text{nxe}} \right\} \tag{45}$$

Then, the calibration loss is just $C_{\text{nxe}} - C_{\text{nxe}}^{\min}$.

[6] This may not hold for a system with significant differences between query-dependent minimum scores, but for the sake of simplicity we will not consider such a case.

[7] The guess is optimistic because for low scores the logarithmic cost function focuses on target trials ($C_{\log}(llr_t) \approx 0$ for non-target trials, whereas $C_{\log}(llr_t) \gg 0$ for target trials), and all the missing trials are assigned the *maximum value* in the set of score values ranging from $-\infty$ to $llr_{\min}(\mathcal{S})$.

4.3 Required amount of processing resources

In NIST 2006 STD [1], systems were expected to have two separate phases: indexing and searching, thus their processing times were reported separately. The indexing time was reported in terms of the so called *Indexing Speed Factor* (ISF): a real-time factor computed as the ratio of the indexing time to the source signal duration. The search time was reported in CPU seconds, meaning the total aggregate time accumulated over all CPUs. The computers used to perform the processing were also described, including key hardware components and the output of a speed calculation program supplied by NIST. Finally, the maximum memory usage was also reported for both the indexing and searching phases.

Systems submitted to Mediaeval 2013 SWS will be also evaluated in terms of the computing hardware, the processing time and the peak memory usage involved in pre-processing and indexing the audio documents and in searching for the queries. We suggest the following protocol (which must be followed for each submitted system):

- The computing hardware used for indexing/pre-processing audio documents and for searching the queries will be described. Separate descriptions will be provided only if different hardware was used in each case. The description will include the number and type of computing nodes. For each type of node, the following information will be provided: computer brand, CPU model (including number of cores and clock speed), RAM capacity and operating system.

- The processing time employed in indexing, or whatever other processing applied to the audio documents before searching, will be reported in terms of ISF, as defined above. The total CPU time will be reported as if it was computed on a single CPU. For instance, if the indexing took 14 hours on a 16-core CPU, the total CPU time would be $14 \times 16 = 224$ hours, and if the total duration of audio documents was 300 hours, then ISF = 0.7467.

- The processing time employed in searching will be reported in terms of the so called *Searching Speed Factor* (SSF): a real-time factor computed as the ratio of the total time employed in processing and searching the set of queries $\mathcal{Q}$ in the set of audio documents Ω to the product of their durations:

$$\mathrm{SSF}(\mathcal{Q}, \Omega) = \frac{T_{\mathrm{Searching}}}{T_{\mathcal{Q}} \cdot T_{\Omega}} \tag{46}$$

  Similarly to ISF, the total CPU time will be reported as if it was computed on a single CPU. For instance, if the searching time was 3 hours on a 16-core CPU, the total CPU time would be $3 \times 16 = 48$ hours. Then, if the total duration of the queries in $\mathcal{Q}$ was 900 seconds (i.e. 0.25 hours) and the total duration of audio documents was 300 hours, it would be SSF = 0.8. If multiple examples $S(q)$ are used to search for a given query q, then $T_{\mathcal{Q}}$ will count the durations of all of them:

$$T_{\mathcal{Q}} = \sum_{\forall q \in \mathcal{Q}} \sum_{\forall j \in S(q)} T_j \tag{47}$$

- The *Peak Memory Usage* (PMU) will be reported separately for the indexing and searching phases, in terms of GigaBytes (GB). If several processes are iteratively executed, the single maximum value for all steps must be reported. If the task is split over several nodes, the single highest value must be reported.

- Finally, for each submitted system a single figure summarizing the required amount of resources, called *Processing Load* (PL), will be computed:

$$\mathrm{PL} = \lambda \cdot \mathrm{ISF} \cdot \mathrm{PMU}_i + (1 - \lambda) \cdot \mathrm{SSF} \cdot \mathrm{PMU}_s \tag{48}$$

  where $\lambda \in [0, 1]$ determines the relative importance of the indexing and searching phases in the evaluation of the required amount of resources. We suggest $\lambda = 0.1$.

References

[1] *The Spoken Term Detection (STD) 2006 Evaluation Plan*, National Institute of Standards and Technology (NIST), September 2006, [Online: http://www.itl.nist.gov/iad/mig/tests/std/2006/].

[2] *The NIST Year 2010 Speaker Recognition Evaluation Plan*, National Institute of Standards and Technology (NIST), April 2010, [Online: http://www.nist.gov/itl/iad/mig/sre10.cfm].

[3] X. Anguera, *Description of 2012 Mediaeval SWS scoring*, MediaEval Benchmarking Initiative for Multimedia Evaluation, August 2012, (Internal Report, distribution restricted to participants).

[4] L. J. Rodríguez-Fuentes, N. Brümmer, M. Penagarikano, A. Varona, M. Diez, and G. Bordel, *The Albayzin 2012 Language Recognition Evaluation Plan (Albayzin 2012 LRE)*, Red Temática de Tecnologías del Habla, June 2012, [Online: https://sites.google.com/site/albayzinlre2012/].

[5] *OpenKWS13 Keyword Search Evaluation Plan*, National Institute of Standards and Technology (NIST), April 2013, [Online: http://www.nist.gov/itl/iad/mig/openkws13.cfm].

[6] *The 2013 Spoken Web Search Task*, MediaEval Benchmarking Initiative for Multimedia Evaluation, June 2013, [Online: http://www.multimediaeval.org/mediaeval2013/sws2013/].

[7] F. Metze, X. Anguera, E. Barnard, M. Davel, and G. Gravier, "The Spoken Web Search Task At MediaEval 2012," in *Proceedings of ICASSP*, Vancouver, Canada, May 26-31, 2013.

[8] N. Brümmer and D. Van Leeuwen, "On calibration of language recognition scores," in *IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, 2006, pp. 1–8.