Description of 2012 Mediaeval SWS scoring

Xavier Anguera

1 Introduction

This short document describes the modifications to the scoring we propose for this year's Mediaeval SWS evaluation. Like in last year's evaluation we use the scripts provided by NIST for the Spoken Term Detection 2006 evaluation [1]. Like last year, we use the ATWV (Actual Term-weighted Value) and the MTWV (Maximum Term-weighted Value) as primary metrics. The difference this year is in the chosen evaluation parameters which aim at treating equally the number of misses and false alarms in the system, regardless of the length of the database being tested or the number of true positives available. Next we will describe the rational behind our changes and give the parameters we will use for scoring this year's and last year's datasets.

2 Evaluation parameters

In this year's Mediaeval SWS evaluation, the primary evaluation metrics are the ATWV (Actual Term-weighted Value) and the MTWV (Maximum Term-weighted Value) as used in the SDT 2006 evaluation. According to [1] and as seen in Equation 1, the Term-weighted value is a function of the miss and false alarm probabilities averaged over all query terms. The "Actual" TWV is obtained by computing the value for a given working point θ and the "Min" TWV is determined by finding automatically finding the θ (working threshold) that obtains optimum results.

$$TWV(\theta) = 1 - \underset{term}{\text{average}}\{P_{miss}(term,\theta) + \beta P_{FA}(term,\theta)\} \quad (1)$$

where θ is the detection threshold and β is the desired relative weight between probability of misses and probability of false alarms, defined as $\beta = C/V(Pr_{term}^{-1} - 1)$, where Pr_{term} is the prior probability of a query term occurring in the reference database and C/V is the cost-value ratio of false alarms. On the one hand, the probability of miss is defined as $P_{miss}(term,\theta) = 1 - N_{correct}(term,\theta)/N_{true}$, where $N_{correct}$ is the number of correct detections of a query term given a threshold θ and N_{true} is the total number of occurrences of that query term. On the other hand, $P_{FA}(term,\theta) = N_{spurious}(term,\theta)/N_{NT}$, where $N_{spurious}$ is the number of incorrect detections of a query term given a threshold θ and N_{NT} is the number of "Non-Target" query term trials, i.e.

Parameter	"Indian Dataset"	"African Dataset"
"similarity"	20s	1s
"find"	20s	0.5s
n_{tps}	1	1
C/V	0.1[1]	β
Pr_{term}	10^{-4}	0.5

Table 1: Parameters used in scoring the "Spoken Web Search" task.

the number of opportunities for incorrect detection. Given that the reference database is a continuous stream defined by duration and not by number of terms/words, N_{NT} is further defined as $N_{NT} = n_{tps}Y_{speech} - N_{true}(term)$ where n_{tps} is defined as the number of trials per second of audio that are conceivable to have in the reference database, and T_{speech} is the length of the database (in seconds).

In addition, the evaluation script defines a "similarity" and "find" parameters. The "similarity" parameter is used when a query term is formed of multiple words by determining the maximum silence allowed between the terms in the reference database to consider them a single reference instance. The "find" parameter indicates how far the mid points of the query term and the reference instance are allowed to still consider a correct detection.

Table 1 shows the parameters used in the 2011 and 2012 evaluations. The parameters for the 2011 evaluation are the same as in the STD 2006 evaluation with the exception of the "similarity" and "find" thresholds that are set to 20 s in order to relax the search constraints given that in the 2011 reference database no forced alignments on the reference data were available and therefore the location of the reference instances were only approximate. In 2012 we obtained forced alignments, therefore these parameters were set more strictly.

While designing the bigger "African Dataset", it was decided that the STD default settings, in which the $P_{FA}(term, \theta)$ is three orders of magnitude more important than $P_{miss}(term, \theta)$ ($\beta = 999.9$), favoring a use case scenario like surveillance, where false alarms are highly penalized, should be adjusted to better reflect the new task. The exact impact of an individual false alarm detection varies with the length of the reference database and the number of query terms used, making the results obtained for the different combinations of reference databases (devel and eval) and query term lists (devel and eval) not fully comparable. In order to address other usage scenarios and to make the different runs comparable we changed the parameters of the system as follows. We rewrite $P_{miss}(term, \theta) = N_{miss}(term, \theta)/N_{true}$ where $N_{miss} = N_{true} - N_{correct}(term, \theta)$ is the number of missed detections of a query term given a threshold. Then, we set the parameters in the system so that for a given ratio between the number of missed terms and the number of false alarms ($N_{spurious}(term, \theta) = \alpha N_{missed}(term, \theta)$) we obtain an equivalent ratio of miss and false alarm proba-

[1]In order to evaluate the "Indian dataset" using 2012's new parameters, the C/V value needs to be set to β as seen below

bilities. In particular, for $\alpha = 1$ we want that $P_{miss} = P_{FA}$ and equal number of misses and false alarms. Other values of α determine different working points. With $\alpha > 1$ we give more importance to missed terms and with $\alpha < 1$ we give more importance to false alarms. Then, if we force $P_{miss} = P_{FA}$ we can find the value of β we need to set as

$$\beta = \frac{n_{tps} T_{speech} - N_{true}}{N_{true}} \tag{2}$$

where n_{tps} can be harmlessly set to 1.0 and T_{speech} and N_{true} depend exclusively on the reference database and the number of query terms available to be detected in each database, and are set a priori and known by the participants. Given the definition of β seen above and the way parameters can be set in the STD scripts, we modified this parameter for each case by setting $Pr_{term} = 0.5$ and $C/V = \beta$, and will require participants to primarily optimize for ATWV, rather than the upper-limit MTWV.

3 Specific evaluation parameters

The listings below are examples how to implement the changes proposed above to force the NIST STDEval.pl script to equally consider misses and false alarms. Only the parameter "-k" needs to be adjusted depending on the different combinations of reference and query datasets. Tables 3 and **??** shows the different values for the Indian and the African datasets.

```
#!/bin/bash
#Scoring script for the indian data: dev data on dev queries
#expects directory with stdlist-file as parameter
export P="../SWS2011/"
export I="../STDEval-0.7/src"
perl -I $I $I/STDEval.pl -s $1/*.stdlist.xml -number-trials-per-sec=1 \
    -e $P/expt_11_std_me2011dev_*_spch_expt_1.ecf.xml \
    -r $P/expt_11_std_me2011dev_*_spch_expt_1.rttm \
    -t $P/expt_11_std_me2011dev_*_spch_expt_1.tlist.xml \
    -A -o $1/score.occ.txt -a $1/score.ali.txt -d $1/score.det -c
        $1/score.cache \
    -S 20 -F 20 -k 13.065 -K 1 -p 0.5 >& $1/score.log
for file in `ls $1/*.plt`
do
    export out=`echo $file|sed 's/.plt/.pdf/'`
    gnuplot $file | ps2pdf - $out
done
```

```
#!/bin/bash
#Scoring script for the African data: dev data on dev queries
#expects directory with stdlist-file as parameter
export P="."
```

Parameter k	Development query terms	Evaluation query terms'
Development dataset	13.06	14.49
Evaluation dataset	12.32	21.59

Table 2: Parameter -k for the Indian data

Parameter k	Development query terms	Evaluation query terms'
Development dataset	15.32	6.36
Evaluation dataset	11.96	19.96

Table 3: Parameter -k for the African data

```
export I="../STDEval-0.7/src"
perl -I $I $I/STDEval.pl -s $1/*.stdlist.xml -number-trials-per-sec=1 \
    -e $P/expt_12_std_me2012dev_*_spch_expt_1.ecf.xml \
    -r $P/afriEval.develop.rttm \
    -t $P/expt_12_std_me2012dev_*_spch_expt_1.tlist.xml \
    -A -o $1/score.occ.txt -a $1/score.ali.txt -d $1/score.det -c
        $1/score.cache \
    -S 1 -F 0.5 -K 1 -k 15.32 -p 0.5 >& $1/score.log
for file in `ls $1/*.plt`
do
    export out=`echo $file|sed 's/.plt/.pdf/'`
    gnuplot $file | ps2pdf - $out
done
```

References

[1] Jonathan Fiscus, Jerome Ajot, John Garofolo, and George Doddington. Results of the 2006 spoken term detection evaluation. In *Proc. SSCS*, Amsterdam; Netherlands, 2007.