Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Fachbereich Informatik

Natural Language Systems Division

Evaluating ASR-Output: Significance testing with SCTK

Ibrahim Dahmash, Kolja Kirsch

Significance testing is crucial to determine whether the results of two different recognizers are just randomly different or indicate a systematic advantage of one over the other. SCTK is a collection of tools that score the output of recognizers (**sclite**), compare those scores (**sc_stats**) and combine the output of different ASRs to a new (and hopefully better) one (**rover**).
We tested the output of three ASRs (Google, Kaldi, Sphinx) on identical input and found that the Google ASR outperforms Kaldi and Sphinx.

SCLITE

sclite aligns the output of an ASR to a given reference file using dynamic programming to perform a global minimization of a Levenshtein distance function which weights the cost of correct words, insertions, deletions and substitiutions [1]. Those alignments can later be used for significance testing with sc_stats and to calculate Sentence and Word Error Rate for the system as well as individual speakers.

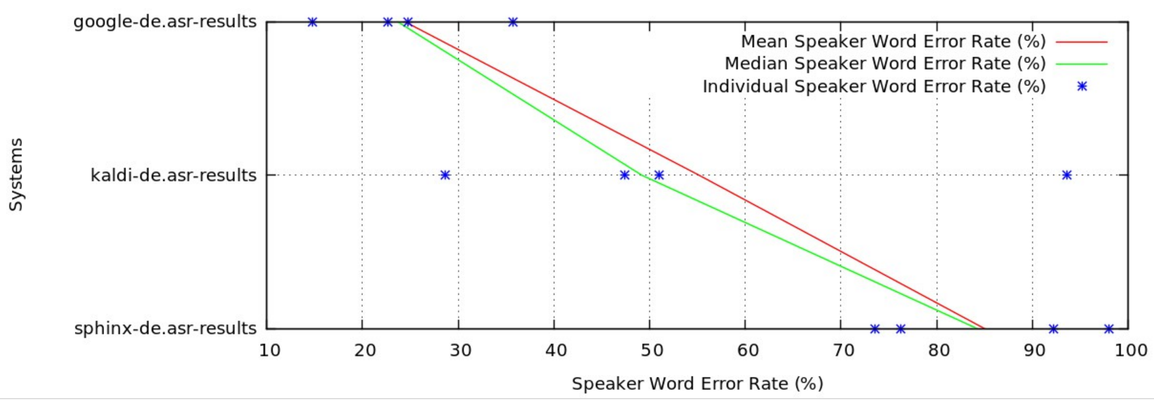
Results for three ASRs

	Google	Kaldi	Sphinx
Sentences with errors	**44.9%**	86.4%	99.5%
Word substitutions	**13.4%**	20.6%	30.9%
Word deletions	**4.3%**	18.3%	7.5%
Word insertions	**2.6%**	3.6%	41.2%
Word Error Rate	**20.3%**	42.5%	79.5%
Word Accuracy	**79.7%**	57.5%	20.5%

Google seems to be the best of the three, to be sure we use the statistical tests that sc_stats provides.

SC_STATS

sc_stats compares the result of multiple ASRs that were given the same test data. With the alignment calculated by sclite we are now able to run different statistical tests. In particular we would like to know if the mean Word Error Rates shown in the following figure are randomly or significantly different (indicating that one system is better than another).



A **Friedman Two-way Analysis of Variance by Ranks** shows that, at the 95% confidence interval, at least one system is significantly different and that the 95% confidence interval the speakers are not significantly different.

A **Sign Test** compares the systems pairwise. To do so it calculates the difference between the WERs of all speakers and then makes a binary decision which speaker was better. Even though all speakers have lower WER in *Google* compared to *Kaldi* the test does not find a difference – the sample size of only four speakers is not big enough to make a decision at the 95% confidence interval. The same applies to the comparisons *Google/Sphinx* and *Sphinx/Kaldi* and the **Wilcoxon Test**.

The **McNemar's** Test [2] compares system A and B by testing if the number of utterances where A was right and B was wrong and those where it was the other way around are randomly distributed.

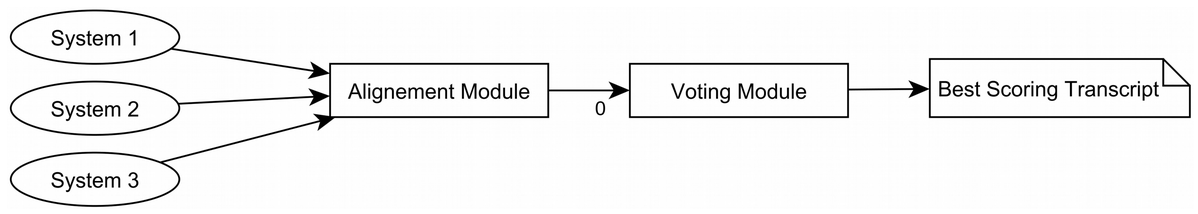
The **Matched Pairs Sentence-Segment Word Error Test** [2] counts the errors in segments of utterances. Because of the samplesize the errors can be assumed to be normally distributed and the mean difference between those normal distributions of errors can be calculated.

Both McNemar and MPSSWE find that Google performed significantly better than Kaldi and Sphinx and Kaldi performed significantly better than Sphinx.

ROVER

Introduction

The Recognizer Output Voting Error Reduction (ROVER) [3] system is used to combine the output of multiple ASRs systems to one output that should have a better quality than each of the base outputs for itself. Rover is implemented in two modules. The first module combines the outputs of multiple ASR systems into a single word transition network (WTN). The second module evaluates the best scoring word using the voting schema.



1. Dynamic Programming alignment

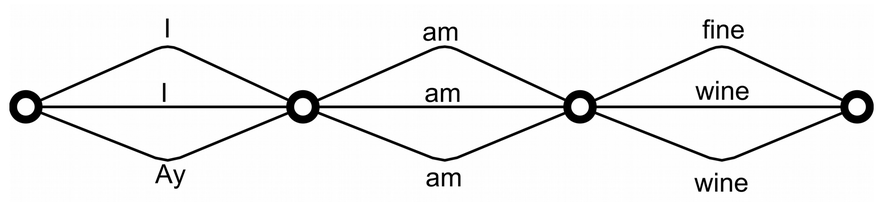
To align more than two WTNs using DP would require a hyper-dimensional search, where each dimension is an input sequence. An approximate solution can be found using the traditional two dimensional DP alignment process (linear alignment) to align result X and Y to a base B and then align B and Z to base B'.

2. Voting Process

Once the composite WTN has been generated from the initial ASR system outputs, the WTN is searched by a voting module to select the best scoring word sequence.

The general scoring formula is $Score(w)=\alpha(N(w,i))+(1-\alpha)C(w,i)$

With a set $W(CS_i)$ of unique word types within a correspondence set CS_i the number of occurences $N(w,i)$ of word type w in CS_i and the confidence score $C(w,i)$, α is the tradeoff between using word frequency and confidence scores.



ROVER implements three voting schemes:

- Frequency of occurrence
  - Ignores the confidence information (α = 1)
- Frequency of occurrence and average word confidence
  - Compute an average confidence score for each word type.
  - Trained on data using a grid-search algorithm
  - Performs better than frequency only
- Frequency of occurrence and maximum confidence
  - Same as average word confidence but with max
  - Similar performance to average word confidence

ROVER requires data in the format

<filename> <waveform channel> <begin time> <duration> word <score>

which we did not have (our format was <word sequence> <utterance ID>) so we did not run ROVER but obviously it would have been epic!

[1] http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm
[2] L. Gillick and S. Cox, "Some Statistical Issues in the Comparison of Speech Recognition Algorithms", ICASSP 89, pp. 532-535.
[3] Fiscus, J. G. (1997, December). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on (pp. 347-354). IEEE.

5dahmash@informatik.uni-hamburg.de, 0kirsch@informatik.uni-hamburg.de
Department of Informatics, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany