Krynica, 14th–18th September 2010

MODIFIED WEIGHTED LEVENSHTEIN DISTANCE IN AUTOMATIC SPEECH RECOGNITION

Bartosz Ziółko, Jakub Gałka, Dawid Skurzok, Tomasz Jadczyk

[1]Department of Electronics, AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków,
{bziolko,jgalka,jadczyk}@agh.edu.pl, skurzok@gmail.com

ABSTRACT

The paper presents modifications of the well know Levenshtein metric. The suggested improvements result in better automatic speech recognition when Levenshtein metric is applied to compare words from a dictionary and speech recognition hypotheses. It allows to evaluate hypotheses and to choose the word which was actually spoken.

INTRODUCTION

An automatic speech recognition (ASR) system needs several layers to work efficiently. One of them is responsible for choosing a word using phoneme hypotheses.

Our acoustic recognition is based on a non-uniform phoneme segmentation and Levenshtein distance [1] (known also as an edit metric) from a sequence of phoneme hypotheses to the phonetic transcription of a word stored in a selected dictionary. This is a part of the system which in this solution is the replacement for word decoder based on hidden Markov model (HMM) [2] frequently used in the standard speech recognition systems [3]. The phoneme segmentation methods are already established and described [4, 5].

The acoustic classifier provides a list of likelihood-ranked phoneme hypotheses with probabilities, for each frame. This ranking is used in the algorithm described in this paper to calculate modified weighted Levenshtein distance. It results in comparing a phone sequence hypothesis and a word from a dictionary. Finally, the N-best list of word hypotheses are chosen.

The paper is organised as follows. The second section describes the standard weighted Levenshtein distance as it is commonly used in ASR. The third section presents our modification and how it can be applied in a word decoder. The fourth section describes the experiment and results on speech recognition. The paper is summed up with conclusions.

LEVENSHTEIN DISTANCE

Levenshtein distance is frequently used in ASR [6, 7]. It measures the number of differences between two sequences of well defined symbols or characters (letters for example). The Levenshtein distance between two sequences is given by the minimum number of operations needed to

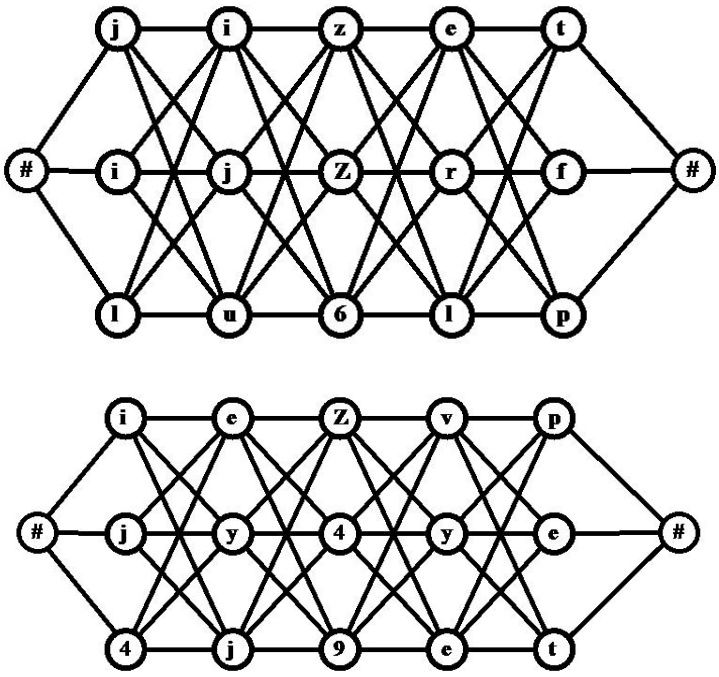

Figure 1. Phonetic hypotheses for words 'Józef' and 'Jerzy. The correct transcriptions are respectively */j u z e f/*' and */j e Z y/*. There are 3 phoneme hypotheses in each column with the most probable one on the top. Errors are easily corrected if secondary and tertiary hypotheses are applied.

transform one sequence into the other, where allowed operations are: insertion, deletion, or substitution of a single symbol. In our case different operations have different weights derived from likelihoods of the characters being modified. Our modification of this measure is described in the next section.

Weighted Levenshtein distance (WLD) between words A and B is

$$WLD(A,B) = \min_{w}\{\alpha r(w) + \beta i(w) + \chi d(w)\}, \quad (1)$$

where α, β and χ are fixed weights, or operation costs and w is a sequence of operations which change A into B, r is the number of replacements, i insertions and d deletions. In ASR, typically A is a hypothesis and B is a phonetic transcription from a dictionary.

SPEECH MODELLING WITH MODIFIED WEIGHTED LEVENSHTEIN DISTANCE

The calculation of the WLD is conducted on phonetic transcriptions (see Fig. 1). A hypothetic sequence from the phoneme classifier is compared with the words taken from a dictionary. The dictionary can be of any finite size.

The values of weights of different operations are a very important detail from application point of view. In our case, they were optimised to maximise the percentage of correct recognitions.

First, Hook-Jeeves optimisation method [8] was applied. However, it resulted in too many local minima. This is why, a much slower, but more accurate method was used based on choosing a grid in the space of possible parameters. Each point of the grid was checked. Then the best point of the grid was chosen as the set of parameters. It did not allow find the global minimum but it allowed to find a set of parameters which are very close to the global minimum. We assumed that

modification cost depends on the obtained data. The weights depend on features and outputs of the classification algorithm.

The acoustic classifier provides a list of best N phoneme hypotheses in each $k = 1..K$ time frame with probabilities p_{nk}, where $n = 1..N$. A substitution cost l_{nk} is higher, if a substituting phoneme is further positioned on the list of other hypotheses. It is then calculated as

$$l_{nk} = \delta \left[\ln(p_{1k}) - \ln(p_{nk})\right], \tag{2}$$

where δ is a parameter. All probability values in the system are implemented as natural logarithms to allow easier computations.

Insertion cost $h_k = -\ln(p_{ins}) = const$ can be described as a cost of performing p_{ins}-probable insertion operation on any k-th position. Probability of such operation can be derived either empirically or from the speech frame versus phoneme rate (undersegmentation rate).

Each deletion cost g can be described as a cost of performing a deletion operation on a k-th segment, classified as a particular phoneme with a maximal probability p_{1k}, where

$$g_k = \ln(p_{1k}) - \ln(p_{del}), \tag{3}$$

and p_{del} is an empirically optimised deletion probability.

It was found experimentally (by checking results for various, different N) that $N = 5$ allows the best possible performance for the given acoustic classifier.

The other parameter, $\delta = 10$, was found using optimisation method. $l_{nk} = 1$ is taken arbitrary if a substituting phoneme is not on the list of N hypotheses (in other words $n > N$).

Then we can present a modified weighted Levenshtein distance (MWLD) as

$$MWLD(A, B) = \min_{w}\{\alpha \sum_{k=1}^{K} l_{nk} r_k(w) + \beta \sum_{k=1}^{K} h_k i(w) + \chi \sum_{k=1}^{K} g_k d(w)\} \tag{4}$$

where $r_k(w) = 0$ if there is no substitution on k-th position and $r_k(w) = 1$ if there is a substitution on k-th position in sequence w of operations to change A into B, and $i(w)$, $d(w)$ define insertions and deletions respectively. Parameters $\alpha = 3$, $\beta = 3$ and $\chi = 1.9$ are weighting functions of overall replacement, insertion and deletion costs.

It is important to maintain proper ratio

$$\beta h_k + \chi g_k > \alpha l_{nk} \tag{5}$$

of these costs, in such a way, that each substitution is more likely than deletion-insertion sequence with the same output and

$$\chi g_k > -\alpha \ln(p_{1k}). \tag{6}$$

EXPERIMENT AND RESULTS

Several experiments were conducted to check various possible sets of parameteres and options. The tests were executed on 100 recordings, different then those used for acoustic classifier training. The speaker was also different, however, a process of speaker adaptation was conducted. Each recording was a complete sentence and phonetic transcriptions of those sentences were used as dictionary entries. For each of it, the mentioned algorithm was applied and several evaluations described in the next section were calculated. Apart from the MLWD, dynamic time warping (DTW) method [2] was used as well and compared.

The results were evaluated in four different ways to compare several parameteres and strategies. The first evaluation criterion is the percentage of correctly recognised sentences. The second one is the percentage of recordings for which the correct sentence is on the list of five strongest

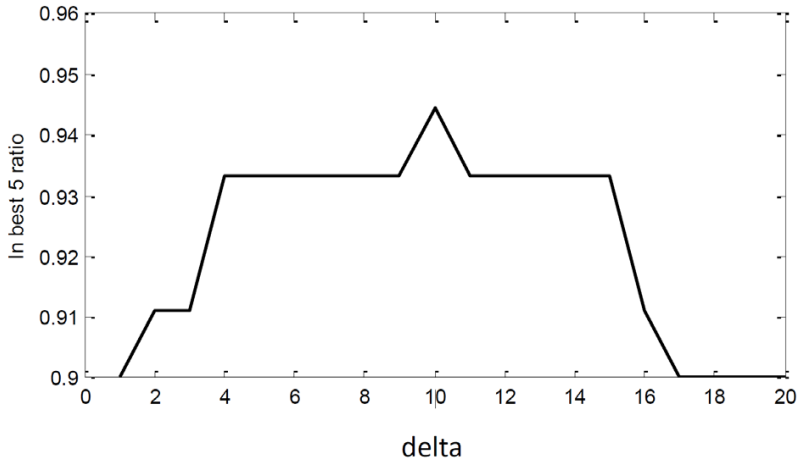

Figure 2. Percentage of correct sentences in the 5 best list of hypotheses depending on the value of δ

hypotheses. The third criterion is the average ranking of the correct sentence on the list of all hypotheses. The fourth one is a distinction factor

$$d_f = \frac{\frac{1}{M}\sum_{m=1}^{M} \ln(p(A = B_m))}{\ln(p(A = B_c))} \tag{7}$$

where M is a size of a dictionary, B_m is an mth word from the dictionary and B_c is a correct recognition and $p(A = B_m)$ is a probability that sequence A is word B_m.

Table 1. Recognition results

method	perfect recognition	in 5-best	averge dist.	distinction factor
DTW	70%	87%	6.7	1.03
MLWD	85%	94%	3.5	1.4

Table 1 shows that MLWD method outperformed DTW in all 4 established evaluation criteria for the test corpus. MLWD is very well tuned to ASR tasks, so the results are not surprising, even though the DTW method is a well balanced and used in applications method as well. Fig. 2 shows the influence of the value of parameter δ on recognition (percentage of test examples for which the correct sentence is in the 5-best list of hypotheses). It points that $\delta = 10$ leads to the best recognition rate. The parameter sets the importance of substitution cost on the position of correct phoneme hypothesis on the list of all phoneme hypotheses of the particular time frame. It is one of the major modification of standard WLD applied by us.

CONCLUSIONS

Presented MLWD is a very good method to compare acoustic hypotheses for speech recognition system with words from a dictionary. It allows to calculate distances between words to maximise the number of correct recognitions. In this way, speech decoding can be conducted with 85% of accuracy on an average dictionary ASR task.

ACKNOWLEDGEMENTS

This work was supported by MNISW grant OR00001905.

REFERENCES

[1] V. I. Levenshtein: *Binary codes capable of correcting deletions, insertions, and reversals*, Soviet Physics Doklady **10** (1966), 707–10.

[2] L. Rabiner and B. -H. Juang: *Fundamentals of speech recognition*, PTR Prentice-Hall, Inc., 1993.

[3] S. Young and G. Evermann and M. Gales and Th. Hain and D. Kershaw and G. Moore and J. Odell and D. Ollason and D. Povey and V. Valtchev and P. Woodland: *HTK Book*, Cambridge University Engineering Department, 2005.

[4] J. Gałka and M. Ziółko: *Wavelets in Speech Segmentation*, Proceedings of The 14th IEEE Mediterranean Electrotechnical Conference MELECON 2008, Ajaccio (2008).

[5] B. Ziółko and S. Manandhar and R. C. Wilson: *Phoneme segmentation of speech*, Proceedings of 18th International Conference on Pattern Recognition (2006).

[6] J. Wu and S. Khudanpur: *Efficient training methods for maximum entropy language modelling*, Proceedings of 6th International Conference on Spoken Language Technologies (ICSLP-00) (2000).

[7] J.-T. Chien and C.-H. Huang and K. Shinoda and S. Furui: *Towards Optimal Bayes Decision for Speech Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP (2006).

[8] C.T. Kelley: *Iterative Methods for Optimization*, SIAM, 1999.