Robust Keyword Search Using Subword Lattice

Andrei Tkachenia, Yan Jingbin, Alexander Trus,
Belarusian State University, Radiophysics dpt., Nezalezhnosti av. 4,
220030 Minsk, Belarus
Tkachenia@gmail.com

Abstract. Keyword search methods based on subword lattice can exclude the problem of Out of Vocabulary Words and improve quality of keyword search. This paper describes the keyword search method based on subword lattice and a-posteriori probability. In the paper it is proposed to use wavelet-based speech feature vector in order to improve the noise robustness of keyword search. The experiments show good keyword search results in noise conditions for the developed algorithm.

Keywords: Keyword search, subword lattice, speech recognition.

1 Introduction

Keyword search in speech files is one of the most difficult tasks in speech data processing. Such technique is necessary in order to do audio indexing, semantic search in audio documents, web search, speech communication check, speech library control [1, 2].

There are several keyword search methods. First and the easiest keyword search method decodes continuous speech into text using automatic recognition system with dictionary and then common text search algorithms apply in order to find keywords for obtained files. The main problem of this method is the limited dictionary. It can not recognize Out of Vocabulary Words, for example: names, acronyms, words from foreign languages. The second keyword search method based on Hidden Markov Model (HMM). It uses HMM for each keyword and single Garbage Model for the remaining words [3]. Sequence of keywords and garbage are formed as a result of speech recognition. But for each new word to be found we need to train not only new HMM for the certain word but also it is necessary to retrain the garbage model. Therefore the usage of this method is very difficult. The latest keyword search method is a subword lattice method. The main idea of this method is that every node of the lattice was matched to a time moment of spoken speech. The main advantage of this method is that it has a good flexibility: even though a phoneme of keyword is not the best hypothesis between nodes of a lattice, anyway it is saved into recognition result. The result of recognition does not depend on keywords set, as we can retrieve any phonemes sequence of requested keyword, therefore in this method we can solve Out of Vocabulary Words problem.

One of the most actual problems of keyword search is the development of noise-resistant algorithms, especially for real-time telecommunication services. In this paper we propose to use wavelet-based speech feature vector in order to improve the noise robustness.

2 Wavelet Based Feature Vector

The Continuous Wavelet Transform (CWT) of $f(t)$ can be defined as:

$$Wf(u,s)=\int_{-\infty}^{+\infty} f(t)\psi_{u,s}(t)dt \ , \qquad (\mathbf{1})$$

where wavelet $\psi -$ function with zero mean, stretch parameter s and shift parameter u:

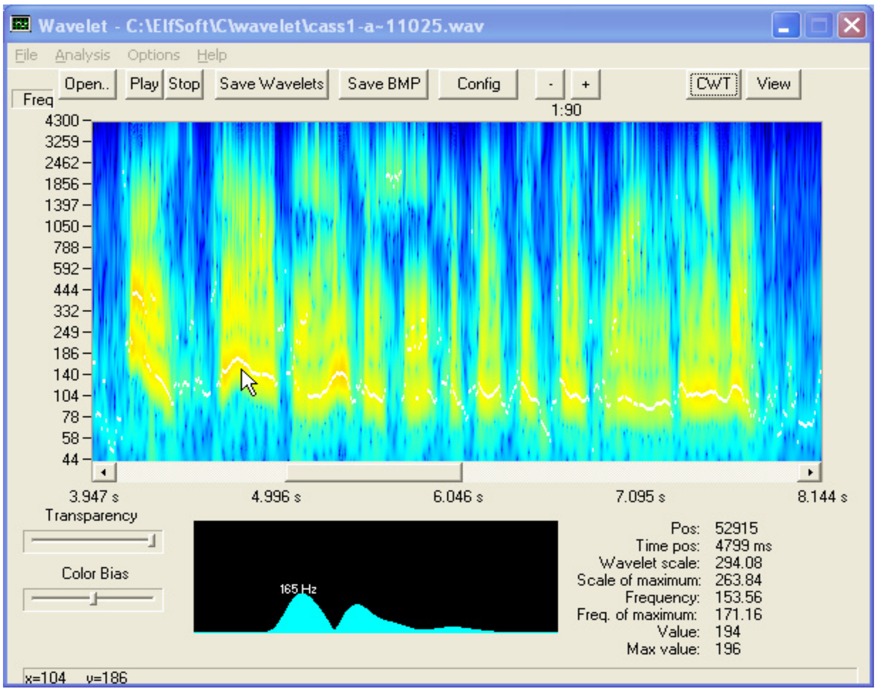
$$\psi_{u,s}=\frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right) \ . \qquad (\mathbf{2})$$

In our work we have used an algorithm for CWT calculation, which implement Morlet wavelet as time-frequency function. Firstly, we used binary version of this algorithm based on powers of 2, to achieve the highest rate. The scale parameter s was changed to $s=2^a 2^{j/J}$, where a – current octave, J – number of voices in an octave. We used $J=8$. Secondly, the pseudo-wavelet was obtained. It combines the averaging power of Fourier transform and accuracy of classical wavelet-transform. We used exponential change of base frequency and linear change of window size. It leads to the full correspondence of frequency scales of wavelet and pseudo-wavelet transforms. In this case (1) transforms to:

$$W_{pseudo}f(u,s)=\int_{-\infty}^{+\infty} f(t)\rho_s(t-u)dt \ , \qquad (\mathbf{3})$$

where $\rho_s(t)$ is a complex pseudo-wavelet with base frequency matched with wavelet frequency in scale s.

The usage of pseudo-wavelets leads to better accuracy for high frequency analysis than it can be achieved using Fast Fourier Transform. Based on wavelet transform (Fig. 1) the cepstrum-like speech parameters were calculated and can be used as feature vectors for keyword search [4].



Wavelet - C:\ElfSoft\C\wavelet\cass1-a~11025.wav
File Analysis Options Help
Open.. Play Stop Save Wavelets Save BMP Config - + CWT View
Freq
1:90
4300
3259
2462
1856
1397
1050
788
592
444
332
249
186
140
104
78
58
44
3.947 s
4.996 s
6.046 s
7.095 s
8.144 s
Transparency
Color Bias
165 Hz

Pos: 52915
Time pos: 4799 ms
Wavelet scale: 294.08
Scale of maximum: 263.84
Frequency: 153.56
Freq. of maximum: 171.16
Value: 194
Max value: 196

x=104 y=186

Fig. 1. Wavelet transform of speech signal.

3 Structure of the Keyword Search System

For the developed keyword search system the lattice-based method was selected as the best one. Lattice $L = (N, A, n_{start}, n_{end})$ is a directional unperiodical graph, where N – set of its nodes, A – set of transitions between nodes and $n_{start}, n_{end} \in N$ – start and end nodes of lattice. Link is represented as $a = (S[a], E[a], I[a], w[a])$, where $S[a], E[a] \in N$ – start and end nodes of edge; $I[a]$ – segment of speech (syllable or phoneme); $w[a] = p_{ac}(a)^{1/\lambda}$ – weight coefficient of link, which is the probability of crossing between nodes; $p_{ac}(a)$ – acoustic likeness; λ – weight coefficient.

For keyword search the following system was proposed (Fig. 2). The search procedure is divided into two steps. First step is the training of recognizer using language and acoustic data base in order to build subword lattice. Second step is the searching of possible keywords and confirming them based on a-posteriori probability.

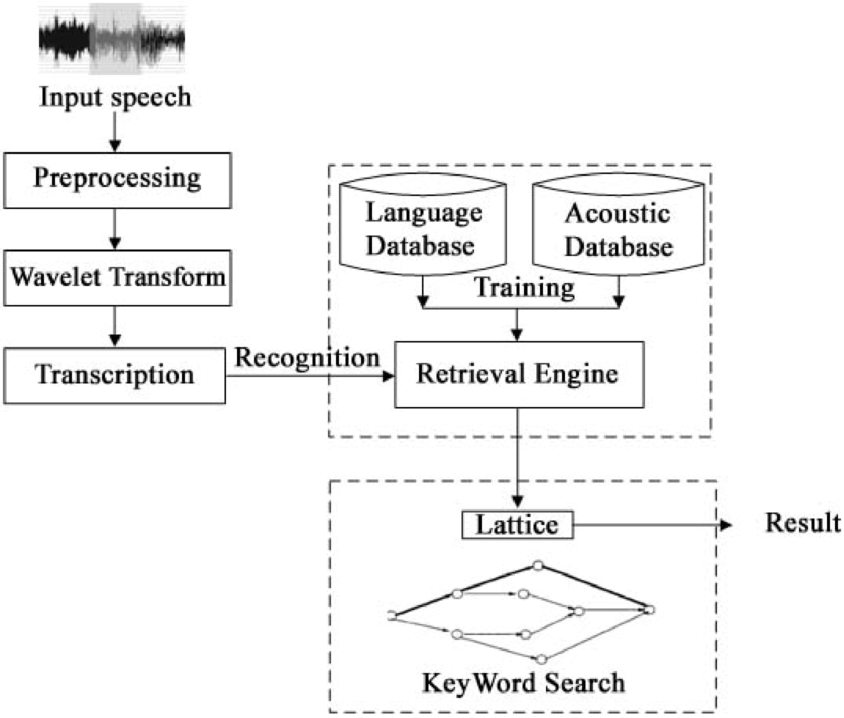


Fig. 2. Keyword search system based on subword lattice.

4 Keyword Search in the Lattice

Supposed that we have syllable sequence $l_1, l_2, ..., l_K$, the sequence of which gives us keyword K_w .

As the result of the search procedure, a set W of the most probable branches w for periodical graph, which corresponds to the given keyword, is obtained. This search is done by finding a path through the lattice structure. Firstly from all nodes of set N we find the start node, which corresponds to the first syllable of a keyword. After that it is necessary to find the next probable node among its linked nodes, which corresponds to the next syllable. This procedure has to repeat K times until we will find the last syllable of a keyword.

After that it is necessary to compute a-posteriori probability for the obtained search result:

$$P(K_w \mid O) = \sum_{\forall w \in W} P(w \mid O) \;, \qquad (4)$$

i.e. it is equal to a sum of a-posteriori probabilities of all possible paths through the graph, which matched with keyword K_w.

A-posteriori probability of the path is computed as:

$$P(w \mid O) = P(O \mid w) \big/ P(O) \;. \qquad (5)$$

For computing $P(w \mid O)$ forward-backward algorithm is used:

1) Let's compute the value of forward probability $\alpha(v)$ and backward probability $\beta(v)$ in accordance with:

$$\alpha(v) = P(O_0^{t(v)} \mid v) = \sum_{w \in W_v^-} P(O_0^{t(v)} \mid w) \;, \qquad (6)$$

$$\beta(v) = P(O_{s(v)}^{T} \mid v) = \sum_{w \in W_v^+} P(O_{s(v)}^{T} \mid w) \big/ b(v) \;. \qquad (7)$$

Based on equations (6), (7):

$$\alpha(v) = \sum_{v_1,\ldots,v_I \in W_v^-} [\prod_{i=1}^{I-1} b(v_i) q(v_i, v_{i+1})] b(v_I) q(v_I, v) b(v) \;, \qquad (8)$$

$$\beta(v) = \sum_{v_1,\ldots,v_I \in W_v^+} q(v, v_1) [\prod_{i=1}^{I} b(v_i) q(v_i, v_{i+1})] \;. \qquad (9)$$

It is obvious that $\alpha(v)$ and $\beta(v)$ can not been directly computed, but they can be defined in recursive form:

$$\alpha(v) = \sum_{(u,v) \in E} \alpha(u) q(u,v) b(v) \;, \qquad (10)$$

$$\beta(v) = \sum_{(u,v) \in E} q(u,v) b(v) \beta(v) \;. \qquad (11)$$

We can compute value of the start $\alpha(v)$ and end $\beta(v)$ nodes:

$$\alpha(v) = b(v)\text{, if } v \in V_0 \ , \tag{12}$$

$$\beta(v) = 1\text{, if } v \in V_T \ . \tag{13}$$

From the start node, value $\alpha(v)$ for each node can be computed and from the end node, we can compute value $\beta(v)$ for each node.

2) Let's compute $P(O)$, which is denominator of equation for computing a-posteriori probability:

$$P(O) = \sum_{w_g \in W_G} P(O,w) = \sum_{v \in V_T} \alpha(v) \ . \tag{14}$$

$P(O)$ can be rapidly computed as the sum of all keywords forward probabilities.

3) As the result we can compute required a-posteriori probability $P(w \mid O)$:

$$P(w \mid O) = \alpha(v_1)[\prod_{k=1}^{K} q(v_k, v_{k+1}) b(v_{k+1})] \frac{\beta(v_K)}{P(O)} \ . \tag{15}$$

The path through the graph which provides the maximum a-posteriori probability forms the found keyword.

5 Experiment

In order to make an experiment the data base of acoustic models was created by using real speech with duration 124 hours. The acoustical models were created for each phoneme based on HMM using HTK toolkit (http://htk.eng.cam.ac.uk). For recognition it was used the recorded speech of «News» broadcast speakers with duration 3.54 hours. Then the data base was processed by special engine in order to mix clear speech with different level noise. There were created two sets of HMMs: acoustical HMMs trained using typical MFCC 39 D feature vector, and HMMs trained using the proposed wavelet-based feature vector. The keyword search based on the created lattice was done. The experimental results for search of 5 keywords in speech with different noise level based on MFCC and wavelet features are shown in Table 1.

Table 1. Keyword search accuracy for noised speech using MFCC and wavelet-based feature vectors.

Noise level, dB	MFCC	Wavelet-based
0	91,2%	91,3%
15	84,5%	88,6%
30	82,9%	85,8%

6 Conclusion

In this paper we have researched the keyword search system based on subword lattice for noised speech. Two types of feature vectors were investigated for this task. Wavelet-based vector shows better results for noised speech than MFCC vector.

References

1. Young S.J., Brown M.G., Foote J.T., Jones G.J.F., Jones K.S. Acoustic indexing for multimedia retrieval and browsing, IEEE International Conference on Acoustics, Speech and Signal Processing (1997)
2. Jones G.J.F., Foote J.T., Jones K.S., and Young S.J. Proceedings of the Workshop “Retrieving spoken documents by combining multiple index sources”, SIGIR, pp. 30-38 (1996)
3. Wilpon J.G., Rabiner L.R., Lee C.H., Goldman E.R. Automatic recognition of keywords in unconstrained speech using Hidden Markov Models, IEEE Transactions on Acoustics, Speech and Signal Processing (1990)
4. Kukharchik P.D., Kheidorov I.E., Martynov D.S., Kotov O.Y. Proceedings of the Workshop “Vocal fold pathology detection using modified wavelet-like features and support vector machines”, EURASIP, Poznan, pp.2215-2218 (2007)
5. Soong F.K., Lo W.K., Nakamura S. Proceedings of the Workshop “Generalized Word A-posteriori Probability (GWPP) for Measuring Reliability of Recognized Words”, SWIM2004, pp. 127-128 (2004)