Research Report A-2007-4

Implementing Speaker Recognition System: from Matlab to Practice

(17.11.2007)

P. Fränti, J. Saastamoinen, I. Kärkkäinen, T. Kinnunen, V. Hautamäki, I. Sidoroff

Speech & Image Processing Unit
Department of Computer Science and Statistics, University of Joensuu
P.O. Box 111, FIN-80101 Joensuu, FINLAND
franti@cs.joensuu.fi

Abstract:

In this paper, we summarize the main achievements made in the 4-years PUMS project during 2003-2007. In addition to the reported research progress, the emphasis is on the practical implementations, how we have moved from Matlab and Praat scripting to C/C++ implemented prototype applications in Windows, Unix, Linux and Symbian environments, with the motivation to enhance technology transfer. We summarize how the baseline methods have been implemented in practice, and how the results are expected to be utilized commercially and otherwise after the project. Brief view for future research challenges is outlined.

During the project, we had two main goals: (1) have a solid baseline that is close to the state-of-the-art, (2) implement this method in all relevant platforms. Besides this, there were no strong exact agenda but all intermediate goals were constructed annually due to the course of progress, reflecting the feedback from our partners, and according to our own understanding what we should do next and what we are capable of. One cannot predict the future and set specific innovative research goals. The only way to reach higher goals is via hard thorough working, but also allowing enough freedom of research along the way to give room for new innovations, which may or may not appear. The project has also been a long learn-by-doing process as well.

Keywords: Speech technology, speaker recognition, voice biometric, forensics research, mobile applications, security applications.

Statistics: 40 pages, 31 figures, 8 tables, 11500 words, 64000 characters.

1. Introduction

This article documents the work done in *speech & image processing unit* (SIPU) during the nationwide 4-years *PUMS*[1] project funded by Tekes[2]. The document covers the history, results, potential applications and future prospects of the research. The project was initiated by Tomi Kinnunen's doctoral studies during 1999-2005, which inspired

[1] Puheteknologian uudet menetelmät ja sovellukset – New methods and applications of speech technology (http://pums.fi)

[2] National Technology Agency of Finland (http://www.tekes.fi)

several projects jointly with other universities, industrial partners, and technology agency joint projects.

At an early stage, the research group participated in an earlier *SUOPUHE*[3] project during 2001-02 as a sub-contractor for University of Helsinki, where prof. Antti Iivonen's group needed tools for automatic evaluation of features that were manually and semi-automatically extracted from speech signal. In its simplest form, the work meant just numerical comparison of feature sets of given two speech samples, but in practice, it turned out to be another case of challenging pattern recognition problem with questions such as how to model the speakers, how to train the models, how to measure dissimilarity, how to deal with mismatch of training and testing conditions, how to combine different feature sets by data fusion, and all the practical aspects that needed to be solved. The earliest programs (*DiscrTest*, *ProfMatch*) originate from this period.

Own project was then initiated in 2002 but the pressure from outside led the group to join forces with other speech technology research groups having similar project plans in closely related fields either in *speech recognition*, *speech synthesis*, or *speech-dialogue* applications, which eventually turned into a large nationwide four years PUMS project. It included all the most important research groups in Finland, several government organizations and companies working in speech technology and its applications. During the first year, the project was coordinated by Tampere university of Technology (prof. Jaakko Astola) but since then university of Turku (prof. Jouni Isoaho) took over the coordinating duty.

Our main focus was on the *speaker recognition* problem (Fig. 1) but some secondary issues also were worked upon, namely *voice activity detection* (VAD), which was seemingly simple but necessary sub-component needed by several partners. They all had their own solutions but no one seemed to be happy with their existing methods, and desired to find a better solution. Thereafter, this problem was studied extensively during the latter stage of the project, and results are reported here as well.

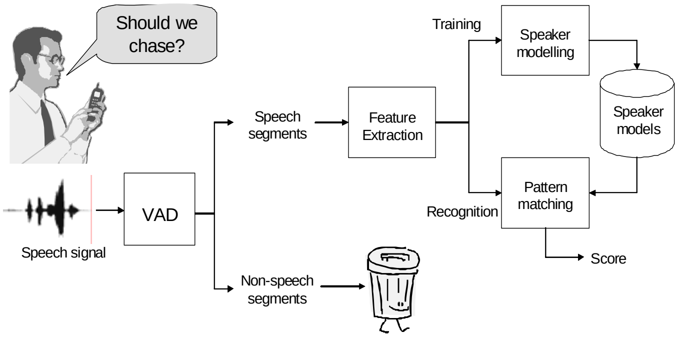


Figure 1: Overall system diagram for speaker recognition.

[3] Finnish Speech Technology: A Multidisciplinary Project (SuoPuhe)

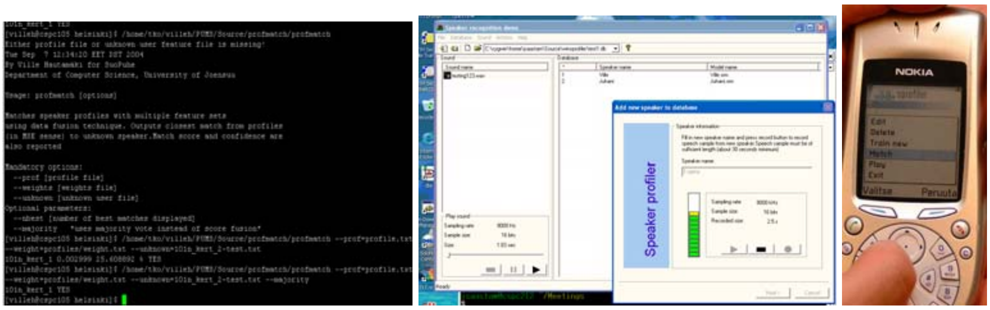
1.1 Speaker recognition history

Despite expectations, the state of art is still based on the same short-term *mel-frequency cepstral coefficients* (MFCCs), features that were invented already in 1980 [Davis'80], augmented by their first and second order derivatives, normalized by cepstral mean subtraction technique [Atal'74, Furui'81], and modeled as *Gaussian mixture models* (GMM) adapted from a so-called *universal background model* (UBM) [Reynolds'00]. The recognition accuracy of the implementation depends a lot on implementation details, proper normalization and training the models in matched acoustic and technical environments, especially avoiding mismatch of channel and other technical factors.

In the beginning, it was unclear how much each factor affects the recognition accuracy, and which processing steps would be vital for successful recognition. During the process, however, these matters were concretely learned by trial-and-error manner within the evolving implementations and during the numerous tests and infamous demonstrations that typically failed 50% of the time when presented in wider public.

At the same time, most of the newest methods were implemented only by Matlab simulations, Praat scripts, and separate C-language components. These lack the capability for being able to make *ad hoc* live demonstrations, larger systematic testing, or provide the methods for project partners or end user as such. The users were not expected to be engineers or computer science professionals, but forensic researchers, police officers, military persons, R&D people at companies utilizing speech technology or developing innovative voice-based systems in completely other fields.

During the project, these methods were implemented step by step in C and C++ languages, and several prototype software systems were built (Fig. 2). These all went through major evaluation steps, and the current versions are Sprofiler 2.3, WinSprofiler 2.3, and EpocSprofiler 2.2.1. The software itself is not made as the result of the project but the systems have been given for free use for the project partners, with the exception of the mobile phone systems after the version 1.0, which haven been copyrighted by NRC and its usage is limited to research only. Therefore, only the EpocSprofiler 1.0 was released within the PUMS project.



```
101n_kert_1 YES
[villeh@cspc105 heleiaki]$ /home/tko/villeh/PUMS/Source/profmatch/profmatch
Either profile file or unknown user feature file is missing!
Tue Sep  7 12:34:20 EET DST 2004
By Ville Hautamaki for SuoPuhe
Department of Computer Science, University of Joensuu

Usage: profmatch [options]

Matches speaker profiles with multiple feature sets
using data fusion technique. Outputs closest match from profiles
(in MSE sense) to unknown speaker.Match score and confidence are
also reported

Mandatory options:
  --prof [profile file]
  --weights [weights file]
  --unknown [unknown user file]
Optional parameters:
  --nbest [number of best matches displayed]
  --majority   "uses majority vote instead of score fusion"
[villeh@cspc105 heleiaki]$ /home/tko/villeh/PUMS/Source/profmatch/profmatch --prof=profile.txt
--weight=profiles/weight.txt --unknown=101n_kert_2-test.txt
101n_kert_1 0.002999 15.408892 4 YES
[villeh@cspc105 heleiaki]$ /home/tko/villeh/PUMS/Source/profmatch/profmatch --prof=profile.txt
--weight=profiles/weight.txt --unknown=101n_kert_2-test.txt --majority
101n_kert_1 YES
[villeh@cspc105 heleiaki]$
```

Figure 2: Prototype applications, in which the developed SRE system has been tested: *Sprofiler* (left), *WinSprofiler* (middle), and *EpocSprofiler* (right).

1.2 Research group in Joensuu

The research was carried on in the *Speech & Image Processing Unit*[4] in University of Joensuu, lead by Prof. Pasi Fränti. The composition and roles of the members in the team changed a lot during the years (Fig. 3). In the early stage, the research issues were mainly solved and supervised by Dr. Tomi Kinnunen even though he spent most of his time in finalizing his thesis, and then two years working in a collaborative institute in Singapore[5] in Dr. Haizhou Li's group[6]. Later due to the course of progress, others started to contribute more to the research development, and the core of the group formed of Dr. Ismo Kärkkäinen, Juhani Saastamoinen and Ville Hautamäki who were all present throughout the entire project and made significant contributions.

In addition, several junior members were recruited and served in the project with various time periods and success, namely Evgeny Karpov, Marko Tuononen, Evgenia Chernenko, Rosa Gonzalez Hautamäki, Radu Timofte, Ilja Sidoroff, and Andrei Oprisan. Evgeny Karpov implemented the first versions of recognition libraries and the first Symbian demonstrator. Marko Tuononen and Evgenia Chernenko were responsible for the VAD development. Rosa Gonzalez Hautamäki assisted, performed the work on F0 feature, and implemented the speech filtering part. Radu Timofte developed and implemented the methods for keyword spotting, and performed the latest VAD development. Ilja Sidoroff and Andrei Oprisan were main contributors for the Windows GUI development.

In addition, Dr. Pavel Kopylov, Victoria Yanulevskaya, Andrei Mihaila, Olga Grebenskaya, Vibhor Jain, Harsh Gupta, Sergey Pauk, Teemu Kilpeläinen, Eeva Pölönen, Timo Viinikka and Pekka Nykänen all contributed to the project either by working in another closely related project, or by completing their MSc thesis related to PUMS. In specific, Pavel Kopylov solved all technical issues for the access control prototype with the help of Harsh Gupta. Victoria Yanulevskaya and Andrei Mihaila worked for the Symbian development.

[4] http://cs.joensuu.fi/sipu/
[5] Institute for Infocomm Research (I^2R)
[6] Speech & Dialogue Processing Lab (http://sdp.i2r.a-star.edu.sg/)

Figure 3: Research group in a meeting in 2005.

1.3 PUMS personnel

The following persons have been employed within PUMS for certain time periods:

Pasi Fränti	Head of the project	03-07
Juhani Saastamoinen	Project manager	03-05 (I, II), 07 (IV)
Ismo Kärkkäinen	Senior researcher (PhD)	07 (IV)
Tomi Kinnunen	Senior researcher (PhD)	07 (IV)
Evgeny Karpov	Project researcher	03-05 (I-II)
Ville Hautamäki	Project researcher	03-06 (I-III)
Marko Tuononen	Project researcher	05-07 (III-IV)
Victoria Yanulevskaya	Project researcher	06 (III)
Evgenia Chernenko	Project researcher	06-07 (IV)
Rosa Gonzalez	Project researcher	05-06 (III)
Ilja Sidoroff	Trainee	06-07 (III-IV)
Radu Timofte	Trainee	06-07 (IV)
Andrei Oprisan	Trainee	07 (IV)

2. Speaker recognition

The method of our baseline uses MFCC features, and *centroid models* (i.e. *vector quantization*) for speaker modeling. Delta features and normalization components were not used at first, simply because they were not needed for the first benchmark data used (TIMIT). Background model was also missing because this was not necessary in speaker identification but it later turned out to be a vital component. After the project ended, the baseline method had been changed to MFCC with its 1st and 2nd order derivatives, utterance level mean/variance normalization, VAD for silence removal, and UBM for background modeling.

The feature set of the baseline implementation remained the same all the way, even though several longer term features were studied, implemented, experimented and ended up into the prototype software as additions. Yet, the baseline method is still composed of the same features and most improvement in the recognition accuracy originated from those additional processing and modeling steps mentioned above.

Biggest catalyst for finding out the critical components and bottlenecks of the method was the participation to *NIST*[7] speaker recognition evaluation (SRE) competition[8] in 2006. From this competition and the latter findings showed interesting results that the simplified variant of the state-of-the-art provides almost the same results with only the carefully fine-tuned baseline method.

2.1. Feature sets used

Our *baseline method* is based on the *mel-frequency cepstral coefficients* (MFCCs), which is a quantized representation of the short-term spectrum (Fig. 4). The audio signal is first divided into 30 ms long frames with 10 ms overlap. Each segment is then converted into spectral domain by *fast Fourier transform* (FFT), filtered and warped according to a psycho-acoustically motivated *mel-scale*, in which lower frequency components are emphasized more than the higher frequency components. Each feature vector consists of 12 magnitudes representing the spectrum after log+DCT conversions, plus the corresponding 1st and 2nd derivatives to model the change and acceleration of the spectrum.

The lowest MFCC coefficient (referred to as C0) represents the log-energy of the frame, and is removed as a form of energy normalization. A two-pass feature normalization, so-called *cepstral-mean subtraction* (CMS) [Atal'74, Furui'81] is then performed for each coefficient to have zero mean and unit variance over the utterance. This is a necessary step and useful for off-line testing. In real-time application, an on-line normalization such as feature warping [Pelecanos'01] or RASTA filtering [Hermansky'94] should be implemented instead. In WinSprofiler 2.3, both of these techniques have been implemented.

The main benefit of using MFCC is that it is the same feature as used in speech recognition, and the same signal processing components can therefore be used for both. This is also its main drawback: the feature tends to capture more speech than speaker

[7] National institute of standards and technology

[8] http://www.nist.gov/speech/tests/spk/2006

related information. If the MFCC features are applied as such, it is a danger that the recognition happens mostly based on the content than on the identity of the speaker. This can be overcome by normalization and background modeling but it affects the matching phase by making it less intuitive and apparently more complicated to implement. Another similar feature, *linear prediction cepstral coefficients* (LPCC), was also implemented and tested but the MFCC remained our choice of practice.

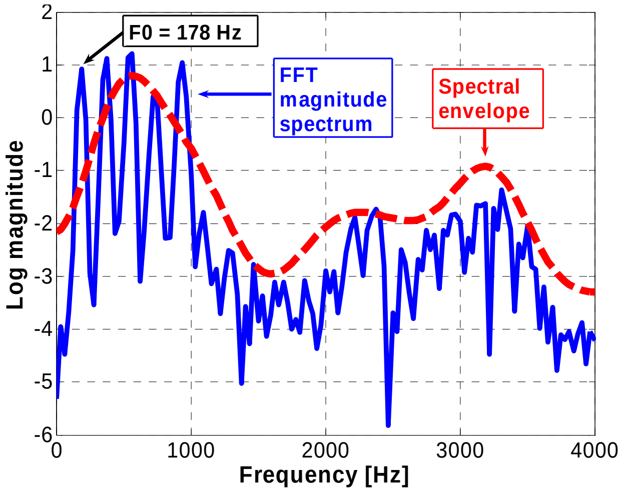


Figure 4. Illustration of a sample spectrum and its approximation by MFCC.

2.2. Longer term features: F0 and LTAS

Besides the short-term features, two longer-term features were studied:

- *Long-term average spectrum* (LTAS)
- *Long-term distribution of the fundamental frequency* (F0)

The first one was motivated by the facts that it includes more details about the spectrum than MFCC, which is filtered to 12 coefficients only; and being average over longer time period it could be more robust on changes in conditions. On the other hand, it was also criticized by the same reasons: it represents only averaged information over time and all information about variance is evidently lost. Moreover, it is not expected to include much more information than is captured in the MFCC representation. However, earlier results suggested that LTAS calculated for /a/ phonemes could provide improvement over MFCC [Kinnunen, Eurospeech'03], and therefore, we decided to study it further.

We had the following research questions and hypotheses for the experiments:

- How does the recognition accuracy of LTAS compare with MFCC?
- How does computational cost of LTAS compare with MFCC?
- Can LTAS and MFCC be fused for improved accuracy?
- Is there any reason to use LTAS in automatic recognition?

Especially the last question was rather strong in our mind as our intuition was that this was expected to be mostly a useless feature. Nevertheless, it was first studied in a student project [Pauk'06], and more detailed later in [Kinnunen'06]. The results (Fig. 5) confirm that the feature is mostly useless, and the following conclusions were drawn:

- Verification accuracy of LTAS: it is much worse than that of MFCC.
- Computational speed of LTAS: it is much faster than MFCC (Table 1).
- Fusion of LTAS and MFCC is not recommended.
- No other reason to use LTAS was found.

To sum up, even though LTAS is used in forensic research for visual examination, its use in automatic analysis has no proven motives. With the exception that as being faster to compute, it could potentially be used as a fast pre-selection tool but so far none of us considered this important enough worth to further studies.

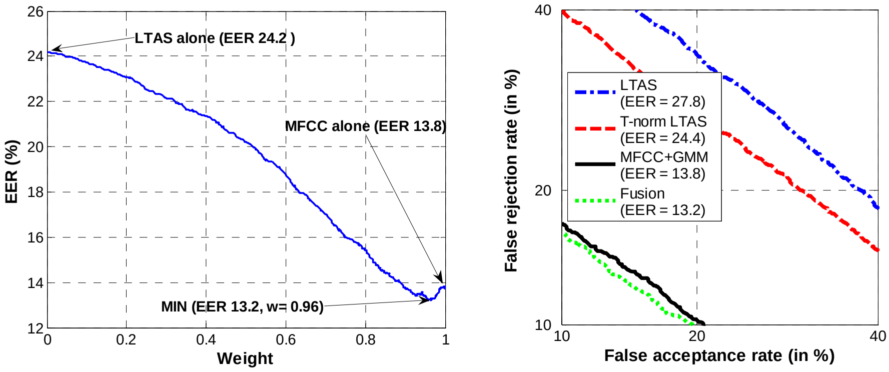


Figure 5: An attempt to improve the baseline by adding LTAS via classifier fusion. The difficulty of tuning the fusion weights is shown left, and the corresponding results of the best combination is shown right for NIST 2001 corpus.

Table 1: Computing times (s) of the different LTAS variants compared to the MFCC.

	Feature extraction	Matching	Total
Single LTAS	0.3±0.1	< 0.01	0.3
Single LTAS + Tnorm	0.3±0.1	1.8±0.2	2.1
Short-term LTAS	0.2±0.1	< 0.01	0.2
Short-term LTAS + Tnorm	0.2±0.1	1.8±0.2	2.0
MFCC+GMM	2.6±0.1	0.6±0.9	3.2

Fundamental frequency, on the other hand, does contain speaker-specific information, which is expected to be independent of the speech content. Since this information is not captured by MFCCs, it can potentially improve recognition accuracy of the baseline system. However, it is not trivial to extract the F0 feature, and how to use it in the matching process. These issues were extensively studied [Gonzalez'05]. At this stage, plain F0 and its histogram model were used but later the method was revised to contain combination of F0, its derivative (delta), and the *log-energy* of the frame

[Kinnunen'05]. This combination is referred to as *prosody vector*, and it was implemented in WinSprofiler 2.0.

The results support that the recognition accuracy of F0 is consistent despite the change of conditions (Fig. 6). In clean conditions, no improvements were obtained in comparison to the MFCC baseline, but the inclusion of F0 improved the results on noisy conditions (additive factory noise with 10 dB SNR) according to our tests. Whether this translates to real-life applications was not verified. In the NIST evaluations (Section 2.3), the effect of F0 is mostly insignificant (or even harmful), probably because the SNR of NIST files is better than the 10 dB noise level in our simulations.

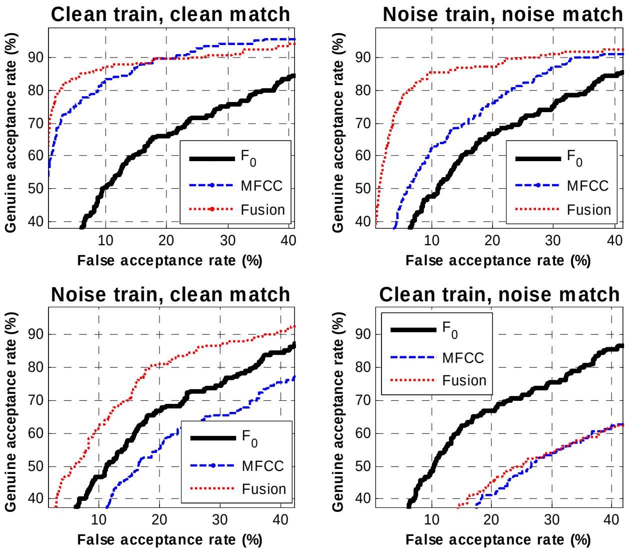


Figure 6: Experimental results supporting noise robustness of F0.

2.3. Speaker modeling and matching

After feature extraction, the problem is to measure the similarity or dissimilarity of a given test sample from the trained models in a speaker database. The question is either from which speaker model the test sample originates from (*identification task*), or whether the sample is close enough to a claimed speaker model (*verification task*).

In the identification task, it is usually enough to find the closest match, or in some applications (e.g. forensic research), find a smaller set (say 3-5) of the best matching speakers for further investigations. In verification, the similarity must be measured relative to a known (or assumed) background model, and draw conclusion whether the sample should be accepted or rejected. A confidence (likelihood) is also desired as well.

Traditional solution is to model the distribution of the feature vectors either by a set of Gaussian mixtures (GMM model), where the sample is clustered, and each cluster is represented by a mean vector, covariance matrix, and a mixture weight (see Fig. 7). A simpler solution is to use only the cluster centroids and assume equal variance. This is often referred to as *vector quantization* (VQ) model because this is what the process essentially resembles.

The choice of the clustering algorithm and model was extensively studied [Kinnunen'08, Hautamäki'08]. We found out that the simpler VQ model provides similar results with significantly simplified implementation. Nevertheless, both methods have been used and implemented in WinSprofiler 2.0, whereas only VQ model and its derivatives (developed later outside of the PUMS project) have been used in EpocSprofiler.

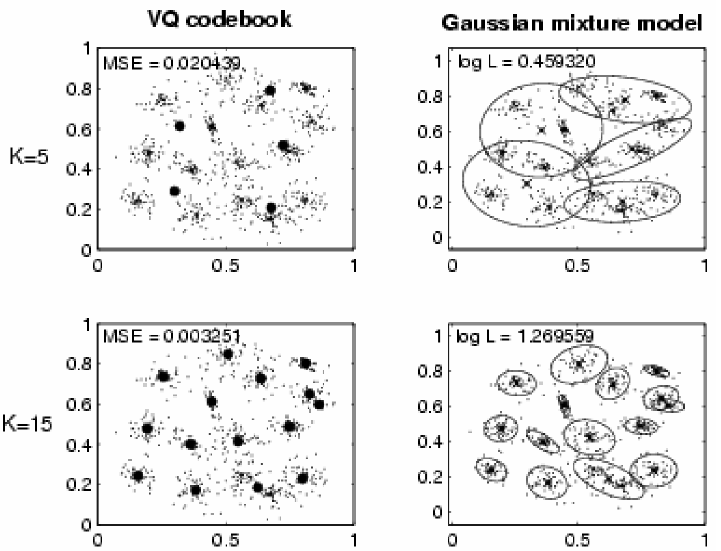


Figure 7: Speaker models by centroids (VQ) and Gaussian mixture model (GMM).

Earlier in the process we mostly focused on the identification task and had overlooked the problems related to the verification task. The NIST evaluations in 2006, however, focused only on the verification task. We suddenly found out that the preliminary results made using VQ-matching were dramatically worse than that of the GMM/UBM model without any apparent reason.

It turned out that the background normalization (UBM) is a crucial component for the success of the verification. Existing solution known as *maximum a posteriori* (MAP) adaptation was originally formulated for the GMM-based model [Reynolds'00]. The essential difference to standard clustering method is that the model is not really trained to match the feature vectors as such, but instead, to model the *difference* from the background model (Fig. 8). Similar solution for the VQ model was then formulated during the project [Hautamäki'08], which solved the training problem.

In principle, the same VQ and GMM modeling approaches and MAP adaptation generalizes to other features such as F0 [Kinnunen'05]. In the case of one-dimensional features such as LTAS, a simpler distance-based approach (Euclidean or Kullback-Leibler) is used. The same problem of adaptation might exist even though rather straightforward *ad hoc* solution was implemented for F0 feature.

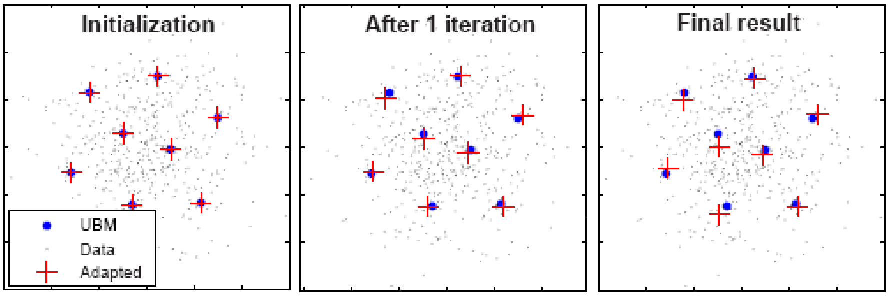


Figure 8: Illustration of the MAP adaptation process.

In addition to modeling a single feature set, a solution is needed to combine the results of independent classifiers. A *linear weighting* scheme optimized using *Fisher's criterion* was first used in [Kinnunen-Interspeech'03] but later, *majority voting* was found out to provide more practical when at least three independent classifiers were present [Kinnunen-SPECOM'04]. Both of these have been implemented in WinSprofiler 2.0 because of their generality, even though the training of the proper fusion weights should be addressed by the user.

More sophisticated solutions were used in NIST evaluations developed by our collaborators at I^{2}R [Tong'06]. These were based on *artificial neural networks* (ANNs) or *support vector machines* (SVMs). General consensus in the follow-up NIST workshop was that fusion itself is needed to achieve the best result, but it is not important which fusion method exactly is used (NN or SVM) but something better than a simple linear weighting was recommended.

On the other hand, it also seems that people tend to avoid fusion in practical solutions because the additional parameter tuning is non-trivial. In this sense, the performance of the method in WinSprofiler 2.0 could be further improved but it is uncertain if it is worth it, or would work in practical application. The use of data fusion is more or less meant for experimental purpose, and not considered as a part of the baseline.

2.4. NIST competition

NIST organizes annually or bi-annually a speaker recognition evaluation (NIST SRE) competition where all interested parties (research group or company) can participate. The organizers have collected speech material and then release part of it as training material, where each sample is labeled by speaker's identity, gender and language spoken. At the time of evaluation, NIST then sends a set of verification trials (about 50.000 in the main category alone) with claimed identity to the participants to analyze. Each participant must send their recognition results (accept or reject claim, and likelihood score) within 2-3 weeks, augmented by a brief documentation of their recognition method used. Each participant is allowed to submit three systems (*primary submission* and two others).

Results were released for the participants and presented in a workshop in June 2006 before the *Speaker Odyssey workshop* [ODYSSEY'06]. Each participant gave presentation of their submitted system, and organizers presented the overall summaries from different subtasks.

We decided to participate when the possibility to send a joint submission with the I^{2}R realized. They had enough manpower necessary for extensive testing, and previous experience on similar language recognition evaluation in 2005 where they were ranked 3rd. This made it realistic to participate since otherwise too large efforts would have been required being away from basic research and development.

In order to avoid overlap, the work load was originally divided so that we focused on F0 and a few experimental features modeled by VQ and histogram models, whereas I^{2}R focused on fusion, GMM-UBM, SVM and ANN models, and some of their own inventions previously used in the language recognition competition [Ma'07]. Both partners had their own implementations of the basic features (MFCC, LPCC). The combinations that showed the best results with previous year NIST corpora were selected.

The main idea was to include three independent classifiers, and calculate overall result by classifier fusion. A variant of the baseline (SVM-LPCC) [Campbell'06] with T-norm [Auckenthaler'00] was one component, F0 another one, and GMM tokenization [Ma'06] the third one (Fig. 9). In this way, different levels of speaker cues are extracted: spectral (SVM-LPCC), prosodic (F0), and high-level (GMM tokenization). Our implementations of GMM and F0 components were used, whereas the SVM and ANN components and the other basic features were provided by I^{2}R. The LPCC feature is based on linear predictive coding (LPC) model, which is a parametric model of the shape. It was chosen since it showed slightly better results than the MFCCs.

Our hypothesis for using F0 was that it could make the system more robust in case the testing data included samples with mismatched acoustic and technical conditions from the training data. However, most of the material was in matched conditions. Furthermore, the results indicate that the threshold learning might become easier, but this was neither confirmed nor disproved by the results. Overall, the effect of F0 was marginal.

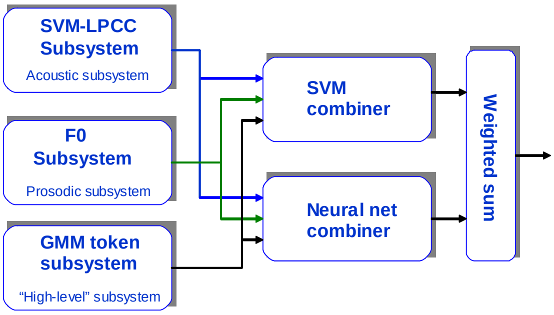


Figure 9: System diagram of the joint submission made by Infocomm at Singapore and University of Joensuu to the NIST speaker recognition evaluation in 2006.

In total, there were 96 submitted systems from the 36 participants from 17 different countries. In the primary task (1conv-1conv), our main submission was ranked 7th out of 36 primary submissions, and our best submission was ranked 16th out of 81 among all submissions (Fig. 10). The corresponding *equal error rate* (EER) was about 7% compared to 4% of the best system [Brummer'07]. However, our method provided 4th best result in the 10sec-10sec test case (least training material) with 21% EER, just after the CRIM submission (17%), and the two I^{2}R solo submissions (21%) among the 24 submissions in this category.

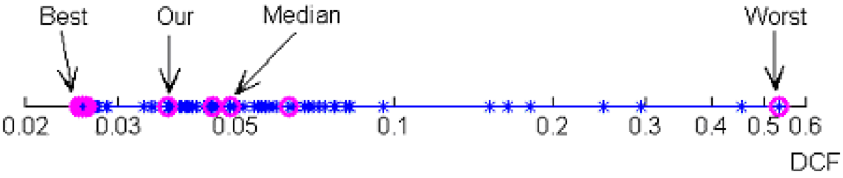


Figure 10: Plot of the results in the primary task (1conv-1conv) according to the DCF cost function (slightly different from EER value) used by NIST.

Our submission did not include revolutionary innovations, and it was merely a combination that worked best according to extensive tests made using corpora of the previous years. The F0 and GMM token subsystems were something that was not used by many, but they did provide improvement when used jointly with the baseline by classifier fusion (see Fig. 11).

At the same time, the method providing the best performance in 1conv-1conv category [Brummer'07] was constructed by a combination of several MFCC-based subsystems similar to ours, combined by SVM-based data fusion. The group at the Brno University of Technology (BUT) reported also simplified variant of their method [Burget'07], showing that similar result can be achieved based on the carefully tuned baseline method without fusion and using multiple sub-systems. Based on analytical comparison with our MFCC baseline, the main components missing are *heteroscedastic linear discriminant analysis* (HLDA) [Kumar'98, Burget'07] and *eigenchannel normalization* [Burget'07]. Besides those, we expect the difference in performance to be mostly due to a tuning of the models and parameters.

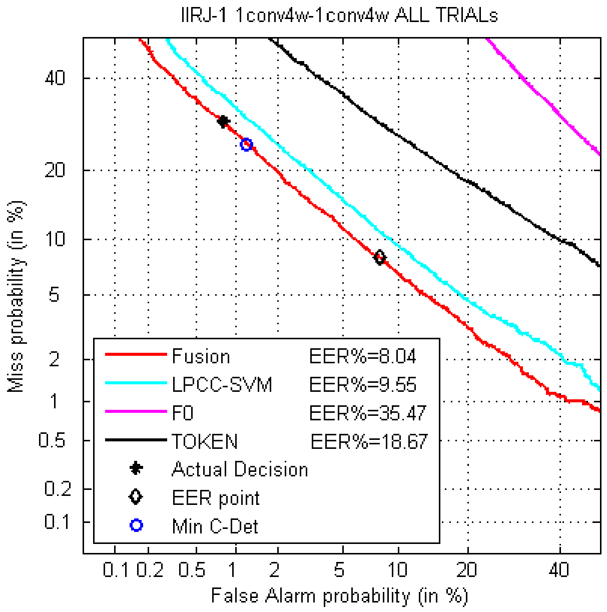


Figure 11: Influence of each component separartely and jointly in our NIST submission.

2.5. Affecting factors

Since the mismatch of training conditions and application environment played an important role, we studied the relative importance of several factors in [Saastamoinen-SPECOM'05]. The factors were divided into three types: technical (T), speech related (S) and data dependent (D) factors. According to the tests made using an early MFCC baseline resulted roughly in the following order of importance of the factors:

1. noise (T)
2. different microphone (T),
3. disguise (S),
4. quality of the sample (T),
5. text reading versus conversational speech (S),
6. sample length (D),
7. language (D),
8. text-dependency (D).

Technical factors were the most crucial for the recognition accuracy, namely noise and the change of microphone. It is noted that the type and amount of noise should be studied further to draw stronger conclusions. Changing of voice (deliberate or subconscious) had also significant effect as the style of reading (text reading vs. spontaneous speech) had a small effect as well. This arises an interesting hypothesis: intruding to a system could be possible by imitation contrary to common expectations.

Technical factors were considered further by studying how Symbian implementation, quality of the microphone in mobile phone, and the GSM coding affect the recognition accuracy. The limitations of the Symbian environment (no floating points) were solved by revising the implementation of the fast Fourier transformation in [Saastamoinen-JASP'05], but the quality of the audio system of mobile phone still weakened the recognition accuracy, as shown in Table 2.

However, it was observed in a student project [Viinikka'04] that the mismatch of conditions (GSM coded or raw speech signal) had the most significant effcet on recognition accuracy. Any coded test sample would match to another coded sample rather than the correct uncoded sample indicating the vulnerability of the system to channel mismatch. A possible solution would be to code all samples via GSM coding but this cannot provide solution for the general channel normalization issue, which remains unsolved.

Table 2: Recognition accuracy for mobile phone environment.

	Avg. recognition rate	std. dev.
FLOAT, Symbian audio	83.2 %	4.38
FLOAT, PC audio	100.0 %	N/A
FIXED, Symbian audio	76.0 %	2.83
FIXED, PC audio	100.0 %	N/A

Data-dependent factors had the least effect in the sense that it did not matter much which phrase was used in the training [Saastamoinen-SPECOM'05]. Utilization of text information itself is two-sided. On one hand, if the language spoken is known, using time-dependent matching (DTW) can improve the accuracy in case of very short (1.9 s) test samples, and when every user had own password [Gupta'05]. Short samples can be the case in real-time access control system but, on the other hand, it is desired that the system would be text-independent.

2.6. Keyword spotting and other tasks

The goal of keyword spotting is to search through audio content based on input queries as text, phonetic transcription or spoken sample, and to return the positions of possible occurrences and the corresponding confidence scores. An easy way to perform word spotting would be to obtain transcription of the speech by *large vocabulary continuous speech recognizers* (LVCSRs), and then performing text retrieval on the transcription. However, this approach does not solve the problem of searching arbitrary keywords from continuous speech.

For the task, we developed a speaker and vocabulary independent method based on so-called *pseudo-phonemes* [Timofte'06]. A prototype of this has been implemented in WinSprofiler 2.3. The method first constructs models for all phonemes in a given language (Finnish and English are supported), and matching scores are computed for each possible location in the audio material. The search is then performed by dynamic programming using the calculated scores as cost function, to find the most probable occurrences. The method in WinSprofiler achieved *equal error rate* (EER) of 3.3 % and *figure of merit* (FOM) of 46 % with TIMIT corpus. A slower improved variant reached the rates of 2.5 % (EER) and 59 % (FOM) for the same corpus [Timofte'07].

Speaker clustering was also studied in [Grebenskaya'05] by clustering speakers into 4 classes (1 female and 3 male classes). Different HMM models were then trained for each class separately, but the results provided only minor improvement and the topic was not studied further.

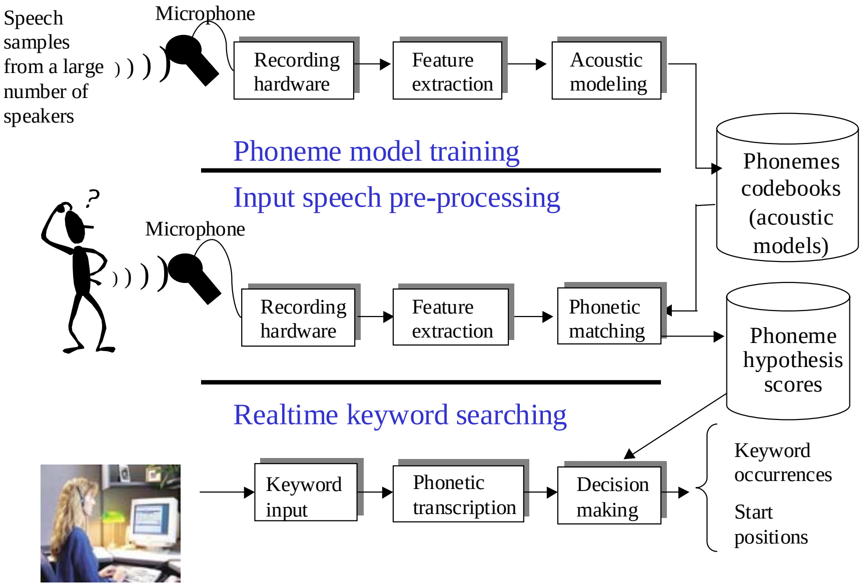


Figure 12: System diagram of the keyword search using so-called pseudo-phonemes.

3. Voice activity detection

During the PUMS project, it became apparent that many of the participants needed *voice activity detection* (VAD), and nobody seemed to be too happy about the existing solutions. The problem is to segment a given input signal into parts that contain speech and the parts that contain background (Fig. 13). This can be done at the frame-level or by combining neighboring frames to achieve longer (e.g. 1 second resolution) segments as the final output. We carried out extensive study of several existing solutions, and developed a few new ones during the course of the project. We considered three different applications.

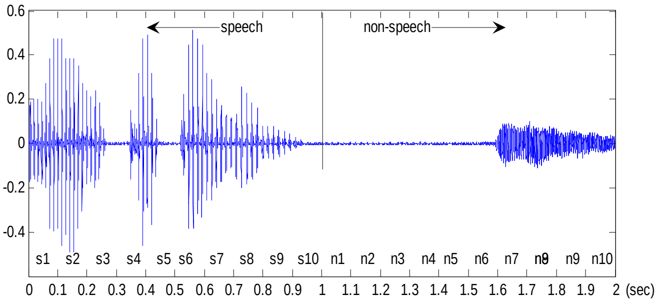


Figure 13: From speech waveform to VAD decisions.

3.1. Applications

In *forensic application* (Fig. 14), there are lots of recordings collected by eavesdropping, and automatic annotating would be needed to save manual work. Biggest challenge is the recording quality. It can vary from a quiet hotel where the microphone can record speech even from the neighbor room, or in a loud music restaurant where it is difficult to even human to recognize what was spoken. Typical audio material is a record of 24 hours a day, and can be days or even weeks long in total.

As an example of interactive voice-based dialogue system, we considered the bus timetable system called *Busman* [Turunen'05]. It provides bus route and timetable information for the city of Tampere, Finland. The user can request certain bus routes such as "*Which bus goes from the Central square to the railway station?*", or timetable information such as "*When does next bus leave from the Central square to Hervanta?*" The purpose of voice activity detector in the system is to detect when the user is speaking, and to extract speech from the input (Fig. 15).

In speaker recognition, we want to model the speaker only from the parts of a recording that contain speech. It is therefore important to use a conservative threshold to make sure that the frames used in the modelling actually do contain speech. If there is lack of speech material, a compromise might need to be taken between having enough training material, and not having too many non-speech frames included. The segmentation can be performed directly at the frame-level.

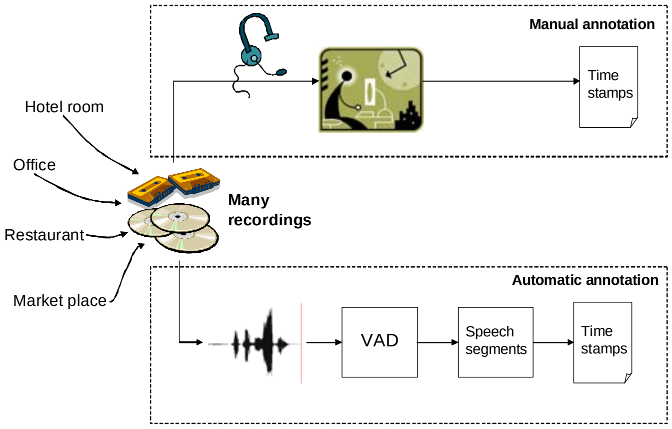


Figure 14: System diagram of forensic skimming application.

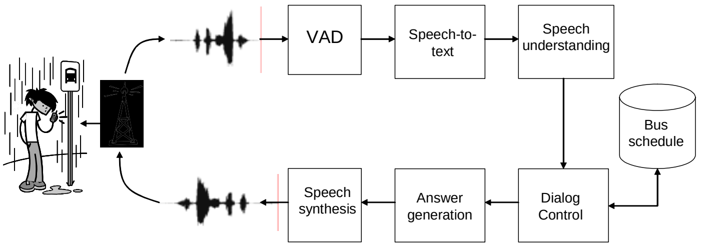


Figure 15: Voice activity detection in the Bus-Stop application.

For evaluation, we use the four data sets summarized in Table 3. NIST 2005 SRE data set is recorded over telephone network and its original purpose is to evaluate speaker verification methods. Only a small subset was selected and manually annotated by the authors. Bus stop data set is recorded over telephone line, and it contains human speech, synthetic speech and DTMF tones. In Lab recording, we simulated forensic eavesdropping application by recording in our laboratory using a hidden distance microphone.

NBI data sets are extracts from a legal eavesdropping recordings made during real criminal investigations by National Bureau of Investigation[9]. Materials are recorded over distant microphone (covert listening device), in extremely challenging conditions, microphone is placed in a location most useful for a forensic investigator. Sometimes microphone is very close to the suspect and in some recordings microphone is not even in the same room.

[9] Keskusrikospoliisi (KRP), http://www.poliisi.fi/krp

Table 3: Speech segmentation data sets.

Material	Recording	Files	Sampling rate (kHz)	Duration per file	Total duration
NIST 2005	Telephone	15	8	5 minutes	1:14:45
Bus-Stop	Telephone	94	8	1.5 – 9 minutes	3:08:13
Lab	Labtec PC mic.	1	44.1	over 4 hours	4:14:42
NBI	Covert listening device	4	16 – 44.1	20 minutes – 2 hours	4:35:47

3.2. Methods

We have experimented both real-time and batch processing variants. Real-time operation is necessary in some VAD applications such as telecommunication and speaker recognition, where latency is an important issue in practice. The application should start to process the extracted feature vectors at the same time when the speaker is still talking. In forensic skimming, on the other hand, real-time operation is not necessary, and the segmentation can be performed as a background process.

VAD methods can also be classified according to whether separate training material is needed (trained) or not (adaptive), see Table 4. Methods that operate without any training are typically based on short-term signal statistics. We consider the following non-trained methods: *Energy*, LTSD, *Periodicity* and the current telecommunication standards: G729B, AMR1 and AMR2.

Trained VAD methods, on the other hand, construct separate model for speech and non-speech based on annotated training data. The methods differ in what features are used, and which modeling method is applied. We consider two methods based on MFCC features (SVM, GMM), and one based on *short-term time series* (STS). All of these methods were developed during the PUMS project. We also modified the LTSD method so that the noise model was adapted from the training material instead of the beginning of the file as in the original method.

Figure 16 shows an example of the process, where the speech waveform is transformed frame by frame to the speech / non-speech decisions using the Periodicity-based method [Hautamäki'07]. First, features of the signal are calculated, and smoothed by taking into account the neighboring frames (five frames in our tests). The final decisions (speech or non-speech) are made according to a user select threshold. In real applications, the problem of selecting the threshold should also be issued. Here we consider only the values of equal error rates, or report the entire operating curve for all possible thresholds.

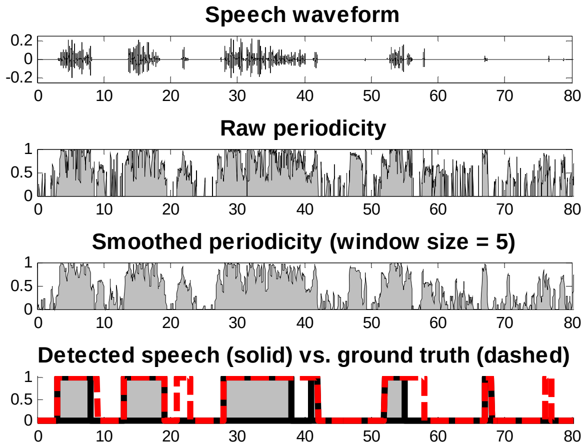


Figure 16: Demonstration of voice activity detection from framewise scores to longer segments using Periodicity method [Hautamäki'07].

Comparisons of the EER results of the different methods are summarized in Table 4, and their performance when varying the acceptance threshold are demonstrated in Figure 17. For G729B, AMR, and STS, we have set the threshold when combining individual framewise decisions to 1-second resolution decisions by counting the proportion of speech and non-speech frames in each segment.

Table 4: EER comparison (%) of the VAD methods with the four data sets.

	VAD method	**NIST 2005**	**Bus stop**	**Lab**	**NBI**
Adaptive	Energy [Tong'06]	1.5	14.6	16.8	30.0
	LTSD [Ramirez'04]	40.0	19.2	14.4	31.8
	Periodicity [Hautamäki'07]	3.2	21.9	9.9	21.4
	G729B [ITU'96]	8.9	6.5	7.9	**13.3**
	AMR1 [ETSI'99]	5.5	5.7	7.2	21.8
	AMR2 [ETSI'99]	8.4	7.4	**5.1**	16.1
Trained	SVM [Kinnunen'07]	11.6	5.2	19.5	---
	GMM [Kay'98]	8.8	7.5	9.7	---
	LTSD [Ramirez'04]	**1.3**	6.2	14.9	---
	STS [Timofte'07]	7.1	**3.9**	8.6	---

For the NIST 2005 data, the simple energy-based and the trained LTSD provide the best results. This is not surprising since the parameters of the method have been optimized for earlier NIST corpuses through extensive testing, and because the energy of the speech and non-speech segments is clearly different in most samples. Moreover, the trained LTSD clearly outperforms its adaptive variant because the noise model initialization failed on some of the NIST files considering speech as non-speech in the beginning, and caused high error values.

For Bus stop data, the energy is not anymore the decisive factor for recognizing speech from background. The developed STS method performs best probably because being able to learn (in training step) the temporal patterns that exist in the samples. Most of the other methods (SVM, GMM, AMR1, G729B) give also reasonable results. Only the methods that rely on energy or periodicity of the signal fail significantly more often.

The NBI data is the most challenging, and all adaptive methods have EER values higher than 10%. The best method is G729B with the error rate of 13%. It is an open question how much better results could be reached if the trained VAD could be used for these data. However, in this case the training protocol and the amount of trained material needed should be studied more closely. Overall, perfect VAD that would work in every recording and environmental condition does not yet exists, according to our experiments.

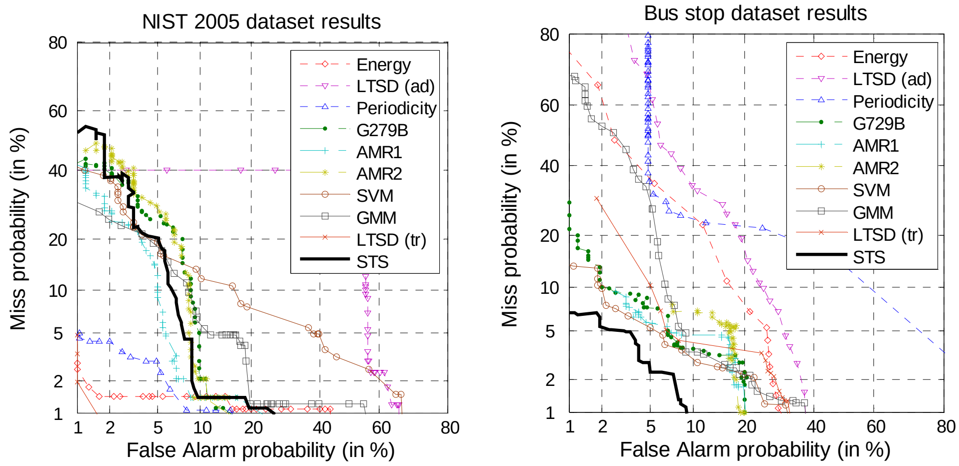


Figure 17: Comparison of the VAD methods for NIST 2005 and Bus stop data.

For WinSprofiler 2.3, we have implemented the three VAD methods that performed best in NIST data: *LTSD*, *Energy* and *Periodicity*. Their effect on speaker verification accuracy is reported in Table 5. The advantage of using VAD in this application with NIST 2006 corpus is obvious, but the choice between *Energy* and *Periodicity* is unclear.

Table 5: Effect of VAD in speaker verification performance (in EER %).

	NIST 2001		NIST 2006
	model size 512	model size 64	Model size 512
No VAD	13.6	16.0	44.4
LTSD	12.4	13.7	35.8
Energy	9.3	10.4	**16.6**
Periodicity	**8.5**	**9.6**	16.8

4. Implementations

Experimentation using Praat and Matlab is rather easy and good for quick testing of new ideas, but not so good for technology transfer and for larger development. In the PUMS project, our aim was to have the baseline methods implemented in C/C++ language for demonstrating the research results, performing large scale tests, and to allow the methods to be tested by the project partners with more critical eyes. Compatibility was also desired in order to allow software integration with the products. External testing by people outside of our research group gave us also a better perspective to usability issues and more practical view to the results.

4.1. Development cycles

Early research was done mostly by Matlab simulations, but the first C-implementations were made by student projects by Ville Hautamäki (matching) and Teemu Kilpeläinen (feature extraction). First *Srlib* library (1.0) was built on the basis of those two and first complete C-language matching program (*Sprofiler*) was implemented in 2002. Research and Matlab experiments were also carried on for classifier fusion as this component was going to be needed later if additional feature sets were going to be used within the prototype software.

At the same time, another software (*ProfMatch*) was implemented during the sub-contracting with the university of Helsinki to perform simple template-based matching and calculate *mean square error* (MSE) scores for pre-calculated features. This was used for testing new experimental features studied in Prof. Iivonen’s group, and extracted using *Praat* and semi-automatic processing. They also implemented their own Windows interface for it using *Tck/Tk*-scripts (see Fig. 18).

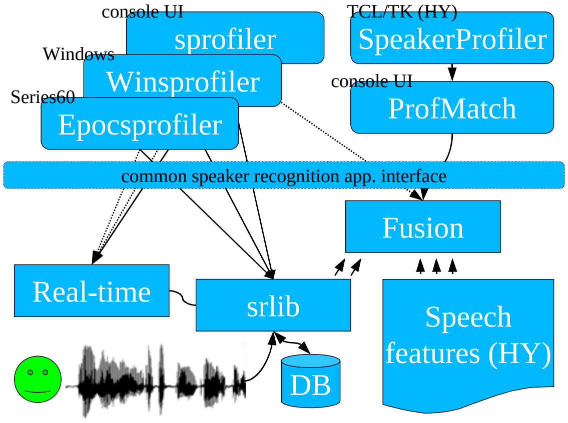


Figure 18: View of the software architecture in 2003.

Meanwhile, we developed Windows prototype using C and C++ languages, first by Evgeny Karpov in late 2003, and the first published version *WinSprofiler* 1.0 was completed in the first half of 2004 by Evgeny Karpov, Olga Grebenskaya and Pavel Kopylov. It was used as the primary testbench whereas we used its command line variant (*Sprofiler*) for all large scale testing. WinSprofiler 1.0 was built on top of an improved speaker recognition library *Srlib2*, which has clear specifications of the functionalities of the training and matching operations, and their input and output data types.

The motivation for *Srlib3* was to develop modularity further and support Symbian in the same library as well, instead of having separate implementation for the mobile and other environments. These were implemented by Juhani Saastamoinen, Andrei Mihaila and Victoria Yanulevskaya. Despite of having working prototypes, too much of the functionalities was mixed with the user interface and all improvements in either part were too difficult to implement in practice with reasonable resources (Fig. 19).

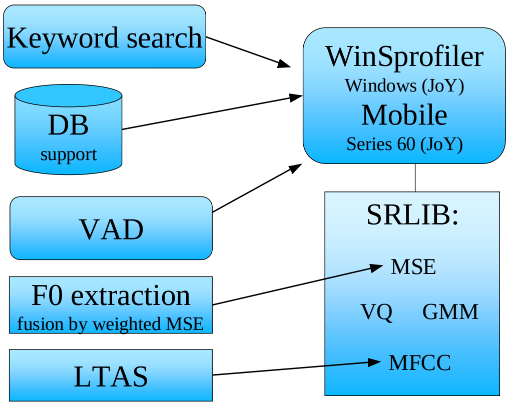


Figure 19: Plans for improvements to be made in 2004.

In order to avoid multiple updates for all software, the library was eventually re-constructed step-by-step as a background work by Juhani Saastamoinen and Ismo Kärkkäinen. This finally ended up to a significant upgrade of the library in 2006 and 2007, which was renamed to PSPS2 (*portable speech processing system 2*), see Fig. 20. Main motivation of this large but invisible work was that the software should be maintainable also after the project would end. To sum up, the following life cycle of the recognition library has appeared during the project: Srlib1 (2003) → Srlib2 (2004) → Srlib3 (2005-2006) → PSPS2 (2006-2007).

As a consequence, all the functionality in WinSprofiler was re-written to support the new architecture of the PSPS2 library so that all unnecessary dependencies between the user interface and the library functionality were finally cleared. This happened as a background project during the last project year (2006-07), and was made mostly by Ilja Sidoroff and Andrei Oprisan. Eventually a new version (WinSprofiler 2.0) was released in Spring 2007, and soon after a series of upgrades were released: 2.1 (June-07) → 2.11 (July-07) → 2.2 (Aug-07) → 2.3 (Oct-07).

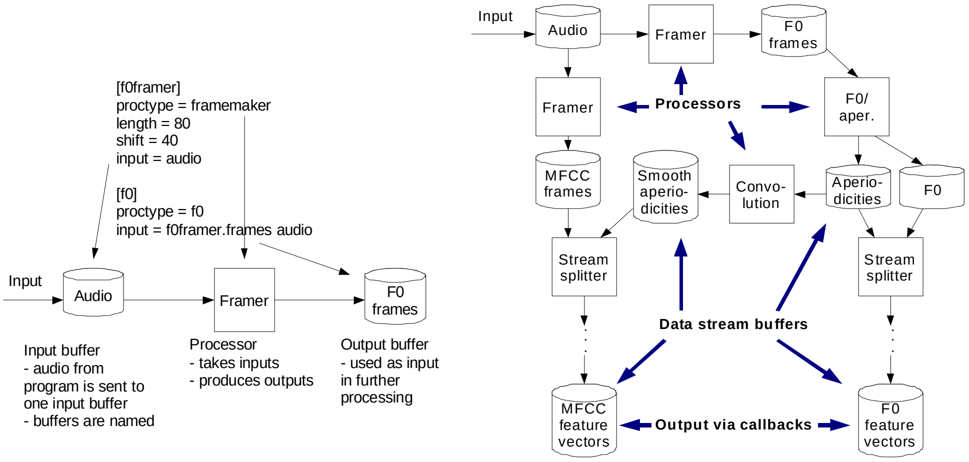


Figure 20: Example of the configurable architecture of PSPS2 using joint MFCC and F0 calculation without any hard coding.

4.2. Design solutions for the implementation

The new version (WinSprofiler 2.0) is written completely using C++ language, consisting of the following components (Fig. 21):

- Database library to handle storage of the speaker profiles.
- Audio processing library to handle feature extraction and speaker modelling.
- Recognition library to handle matching feature streams against speaker models.
- Configurable audio processing and recognition components.
- Graphical user interface.

The GUI part is based on 3rd party C++ development libraries, *wxWidgets*. Existing libraries were also used for the audio component: *libsndfile* and *portaudio*. Database support was implemented using *SQLite3*. All the rest was then implemented by us: signal processing, speaker modeling, matching and graphical user interface. The new version was extensively tested, and the functioning of the recognition components was verified step-by-step with the old version (WinSprofiler 1.0). The new library architecture is show in Fig. 21, and the internal class structure in Fig. 22.

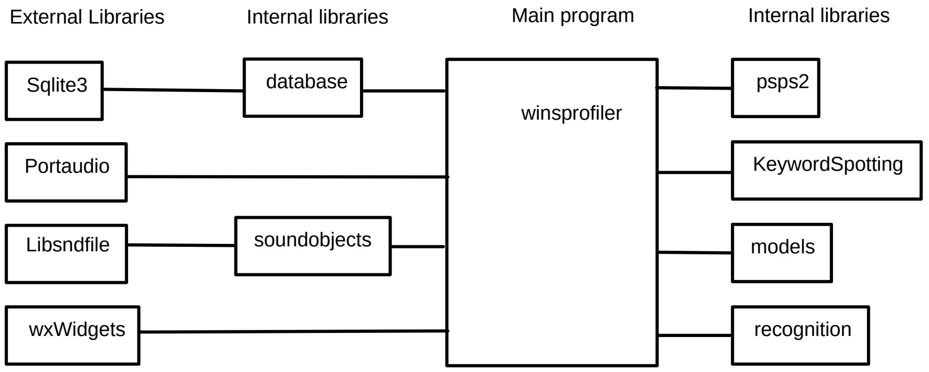


Figure 21: Technical organization of the WinSprofiler 2.0 software.

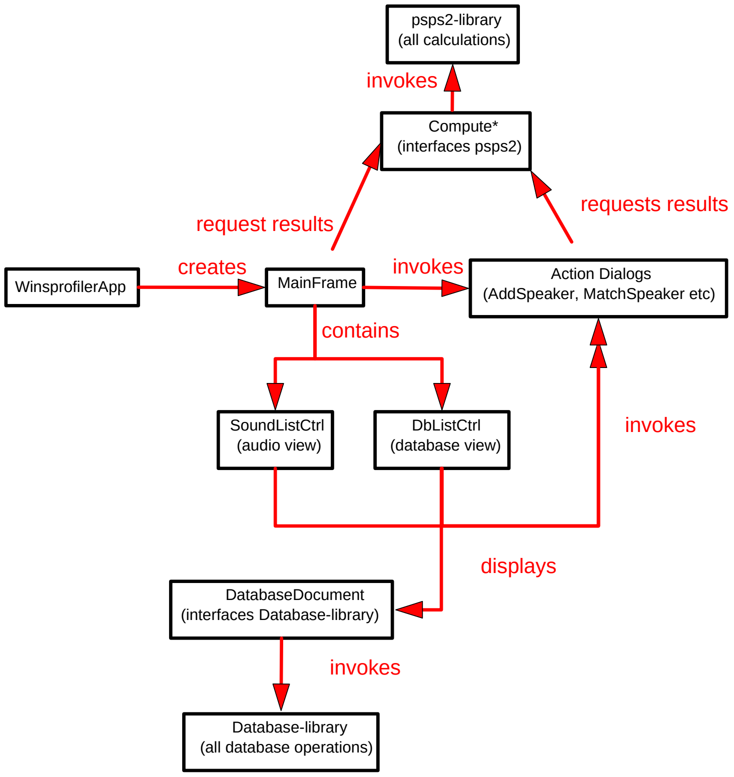


Figure 22: Internal class structure of WinSprofiler 2.0.

4.3. Symbian implementation

During the first project year, the development of a *Symbian* implementation was also started with the motivation to implement a demo application for Nokia *S60 series* phones. Research was carried on for faster matching techniques by speaker pruning, quantization and faster search structures [Kinnunen'06]. The existing baseline (Srlib 2.0) was converted to Symbian environment in order to have real-time signal processing and MFCC feature extraction including instant on-device training, identification, and text-independent verification from spoken voice samples.

The development was made co-operatively with Nokia Research Center during the first project year, and the first version (*EpocSprofiler* 1.0) was published in April 2004. The Symbian development was then separated from PUMS and further versions of the software (EpocSprofiler 2.0) were developed separately, although within the same research group, using the same core library code, and mostly by the same people.

The main challenge was that the CPU was limited to fixed-point arithmetic. Conversion from float-point to fixed-point itself was rather straightforward but the accuracy of the fixed-point MFCC was insufficient. Improved version was developed [Saastamoinen-JASP'05] by fine-tuned intermediate signal scaling, and more accurate 22/10 bit allocation scheme of the FFT, as illustrated in Fig. 23.

Two voice model types were implemented: centroid model with MSE-based matching as the baseline and a new much faster experimental modelling method called *background cell histogram model* and entropy-based matching was developed for EpocSprofiler 2.0 (report is under progress). In identification, training and recognition response of the new histogram models is about 1 second on a database of 45 speakers, whereas the training and identification of the older baseline method was more than 100 times slower. In verification, the recently developed background model called VQ-UBM [Hautamäki'08] was also utilized in the implementation.

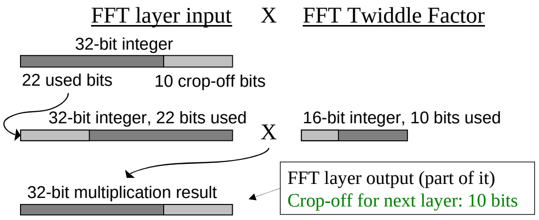

FFT layer input
X
FFT Twiddle Factor
32-bit integer
22 used bits
10 crop-off bits
32-bit integer, 22 bits used
X
16-bit integer, 10 bits used
32-bit multiplication result
FFT layer output (part of it)
Crop-off for next layer: 10 bits

Figure 23: Developed information preserving FFT (22/10 scheme) for Symbian.

4.4. Access control demonstrators

For access control demos, a software called *DoorSprofiler* was developed in 2005 on the basis of WinSprofiler 1.0. It is compatible with *Securitas*' SOAP system and *ESMI*'s door control systems. Real working prototype was implemented in SIPU laboratory in Joensuu Science Park using ESMI door controller unit and a simple short-cut to a door opening relay, in order to avoid any interfering to the access control unit used in the rest of the house, see Fig. 24.

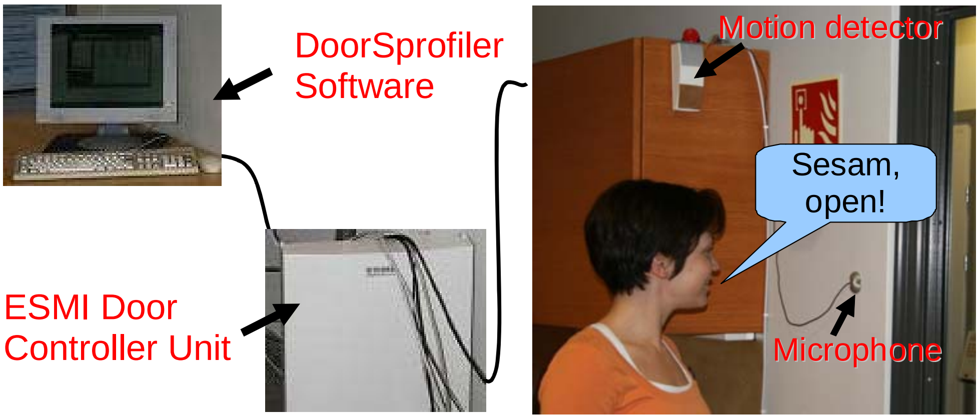


Figure 24: Example of DoorSprofiler in action.

A support for *Kone*'s elevator *OPC server* control system was designed and implemented for *LiftSprofiler* software, where a two-class classification was designed: *staff* and *non-staff* users. Verification threshold was set to have low false rejection (FR) rate so that staff could always enter the lift conveniently. Letting a few non-staff people to use the lift is not considered harmful: false acceptance (FA) rate is not so critical.

A real demonstrator was designed for an elevator in Science Park. Even though it has not yet been installed in real-life, all implementations to make such installation in practice has been made for the scenario shown in Fig. 25.

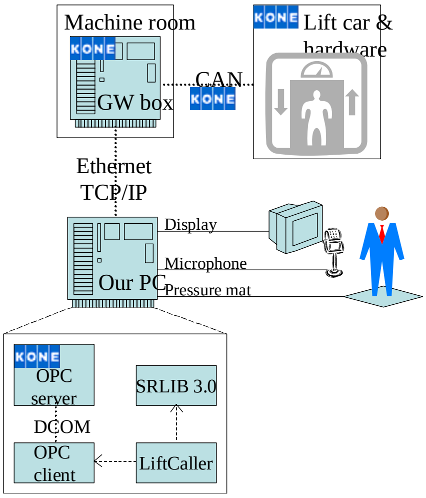


Figure 25: Designed system for implementing voice-based elevator calling.

5. Summary of the main results

The main results of the project can be summarized in the three demonstrators:

- WinSprofiler demonstrator (Windows)
- Mobile phone demonstrator (Nokia Series S60)
- Door access control demonstrator in Joensuu Science Park.

The first prototype supports the following operations:

- Speaker modeling and matching (batch processing and real-time recognition)
- Keyword spotting (support for Finnish and English languages)
- Voice activity detection (also in *VoiceGrep* software)
- Digital filtering for band elimination.

Some of the software (namely Srlib 1.0, WinSprofiler 1.0, EpocSprofiler 1.0, VoiceGrep 0.2 software) are available as project results, whereas the others are copyrighted by the University of Joensuu, and available only as binaries. Other results of the project include the publications and theses listed in the end of this report.

5.1. Recognition results

Even though usability and compatibility are important issues for a practical application, an important question for voice-based user authentication to be accepted into real application, is the identification and verification accuracy the system can provide. We have therefore collected here the main recognition results of the methods developed during the project, and made an attempt to compare them with the state-of-the-art (according to NIST evaluation), and provide indicative results from comparisons with existing commercial programs.

Table 6: Databases that have been used in the evaluation.

Corpus	Trials	Speakers	Length of training data	Length of test data
NIST 2001 (core test)	22,418	174	2 min	2-60 s
NIST 2006 (core test)	53,966	731	5 min	5 min
Sepemco	494	45	12-60 s	9-60 s
TIMIT	184,900	430	15-35 s	5-15 s
NBI data	62	62	42-150 s	10-93 s

We have also used (or considered) *TIMIT*, *Helsinki corpus*, *OGI* and Estonian *SpeechDAT* corpora earlier in the project. However, once we achieved 0% error rates for the first two corpora, we have limited all the large-scale testing for the NIST and Sepemco databases. Other NIST corpora have also been used occasionally, namely NIST 1999, 2002, 2004 and 2005, or a smaller subset for reducing the processing time. The results for the NBI databases have been provided by Tuija Niemi-Laitinen at the Crime laboratory in National Bureau of Investigation, Finland.

The following versions have been included here:

- WinSprofiler 1.0: An early demo version from 2005 using only the raw MFCC coefficients without deltas, normalization, and VAD. VQ model of size 64 is used. This version has also been used in DoorSprofiler version 1.0 and the door demo.
- WinSprofiler 2.0: A completely new version that was released in May 2007, although the recognition library PSPS2 was developed already in late 2006. Main differences were use of GMM-UBM, deltas, and normalization. The first version did use neither VAD nor gender information (specific for NIST corpus).
- WinSprofiler 2.11: Version released in June 2007, now included gender information (optional) and several VADs, of which the periodicity-based method [Hautamäki'07] has been used for testing. The newer version 2.3 includes also a real-time matching and audio filtering, but has the same recognition method.
- EpocSprofiler 2.1: Symbian version from October 2006. Corresponds to WinSprofiler 1.0 except that the histogram models are used instead of VQ.
- IIRJ: The joint submission to NIST competition based on the LPCC-SVM, GMM tokenization and F0 features, and fusion by NN and SVM. Energy-based VAD. This system does not exist as a program, as the results have been constructed manually from the results of several scripts.
- NIST state-of-the-art: The results released by the authors providing the winning method in NIST competition as a reference.

The main results (verification accuracy) are summarized in Table 7 as far as available. The challenging NIST 2001 corpus has been used as the main benchmark since summer 2006. Most remarkable lesson is that, even though the results were reasonable for the easier datasets (TIMIT), they are devastating for *WinSprofiler* 1.0 when NIST 2006 was used. The most remarkable improvements have been achieved in the latter stage of the project since the release of the PSPS2 library used in *WinSprofiler* 2.11. The overall development in recognition accuracy during the project is also visualized in Fig. 26 by using the detection error trade-off (DET) plots.

Another observation is that the role of VAD was shown to be critical for NIST 2006 evaluation (45% vs. 17%), but this did not generalize to *Sepemco* data (7% vs. 13%). This arises the questions whether the database could be too specific, and how much the length of training material would change the design choices and parameters used (model sizes, use of VAD). Although NIST 2006 has a large number of speakers and huge amount of test samples, the length of the samples is typically long (5 minutes). Moreover, the speech samples are usually easy to differentiate from background by a simple energy-based VAD. The background noise level is also rather low.

Table 7: Summary of verification (equal error rate) results (0 % is best) using the NIST 2001, NIST 2006 and the *Sepemco* database.

Version	Sepemco	TIMIT	NIST 2001	NIST 2006 10sec	NIST 2006 1conv
EpocSprofiler 2.1 (2006)	12 %	8 %	---	48 %	46 %
WinSprofiler 1.0 (2005)	24 %	---	33 %	43 %	48 %
WinSprofiler 2.0 (no-vad)	7 %	3 %	16 %	40 %	45 %
WinSprofiler 2.11 (2007)	13 %	9 %	11 %	31 %	17 %
NIST submission (IIRJ)	---	---	---	22 %	7 %
State-of-art [Brummer'07]	---	---	---	---	4 %

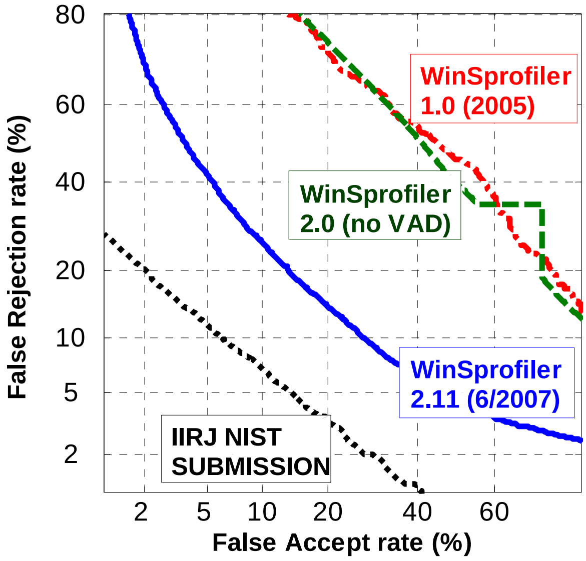


Figure 26: Development of the recognition accuracy during the project (NIST 2006).

5.2. Comparisons with NBI data

Speaker identification comparisons with three selected commercial software (*ASIS*, *FreeSpeech*, *VoiceNet*) are summarized in Table 8 using NBI material obtained by phone tapping (with permission). Earlier results with WinSprofiler 1.0 for different dataset have been reported in [Niemi-Laitinen'05]. The current data (TAP) included two samples from 62 male speakers: the longer sample was used for model training and the shorter one for testing. The following software have been tested:

- WinSprofiler, Univ. of Joensuu, Finland, www.cs.joensuu.fi/sipu/
- ASIS, Agnitio, Spain, http://www.agnitio.es
- FreeSpeech, PerSay, Israel, http://www.persay.com
- VoiceNet, Speech Technology Center, Russia, http://www.speechpro.com
- Batvox, Agnitio, Spain, http://www.agnitio.es

The results are summarized as how many times the correct speaker is found as the first match, and how many times among the top-5 in the ranking. WinSprofiler 2.11

performed well in the comparison, which indicates that it is at par with the commercial software (Table 8).

Besides the recognition accuracy, *WinSprofiler* was highlighted as having good usability in the NBI tests, especially due to its ease of use, fast processing, and the capability to add multiple speakers into the database in one run. Improvements could be made for more user-friendly processing and analysis of the output score list though.

Overall, the results indicated that there is large gap between the recognition accuracy obtained by the latest methods in research, and the accuracy obtained by available software (commercially or via the project). In NIST 2006 benchmarking, accuracy of about 4 to 7% could be reached by the state-or-the-art methods such as in [Bummer'07], and by our own submission (IIRJ).

Direct comparisons to our software WinSprofiler 2.11, and indirect comparisons to the commercial software gave us indications of how much is the difference between "*what is*" (commercial software, our prototype) and "*what could be*". It demonstrates the fast development of the research in this area, but also shows the problem that tuning towards one data can set lead undesired results for another data set.

Table 8: Recognition accuracies (100% is best) of WinSprofiler 2.11 and several commercial software for NBI data (TAP).

Software	Used Samples	Failed samples	Top-1	Top-5
ASIS	51	11	67 %	92 %
WinSprofiler 2.11 (*)	51	11	53 %	100 %
WinSprofiler 2.11	62	0	53 %	98 %
FreeSpeech	61	1	74 %	98 %
VoiceNet	38	24	29 %	52 %

(*) Selected sub-test with those 51 samples accepted by ASIS.

6. Applications

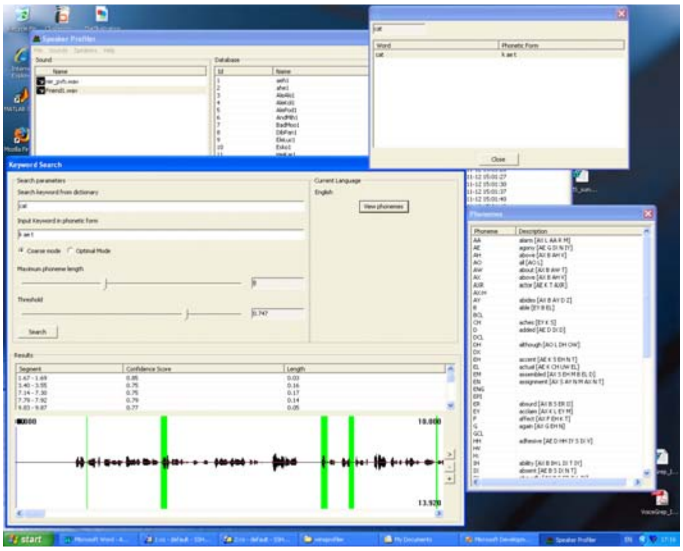
Voice-based recognition is not mature enough to be used as such for person identification in access control. It is applicable mainly in situations where traditional methods (key, RFID, passwords) or other biometrics (fingerprints, iris) are not available, and when user-convenience is preferred over security. Influence of background noise and changes in conditions affect too much the recognition accuracy.

The methods, however, can be useful in certain niche applications as such. In the following, we list potential applications roughly in the order in which the methods are currently usable:

- *forensic research* (supportive tool, useful already) [Niemi-Laitinen'06]
- *border control* (additional security, could be implemented in near-future)
- *internet banking* (additional security, technically possible)
- *call-centers* (cost savings, requires improved quality)
- *helpdesks* (additional security / cost saving, useful for the first motive)
- *tele-conferencing* (speaker segmentation, potentially useful)
- *audio mining* (speaker diarization, potentially useful)
- *access control* (non-critical scenarios, high technical challenges)

In forensic research, any additional piece of information can guide the inspections to the correct tracks. Even if 100% matching cannot be reached, it can be enough to detect the correct suspect high in ranking. Augmented with keyword spotting, voice activity detection and audio filtering, software such as *WinSprofiler* can serve as a practical tool (Fig. 27). The final version of the software supports the following:

- Speaker recognition and audio processing.
- Speaker profiles in database.
- Several models per speaker.
- Digital filtering of audio files (version 2.3).
- MFCC, F0 + energy and LTAS features.
- GMM and VQ models (with and w/o UBM)
- Voice activity detection by energy, LTSD and periodicity-based methods.
- Keyword search (support for Finnish and English languages).
- Fully portable (Windows, Linux and potentially Mac OS X).



Keyword Search

Search parameters

Search keyword from dictionary: cat

Input Keyword in phonetic form: k ae t

● Coarse mode ○ Optimal Mode

Maximum phoneme length: 8

Threshold: 0.747

Current Language: English

Results

Segment	Confidence Score	Length
1.67 - 1.69	0.85	0.03
3.40 - 3.55	0.75	0.16
7.14 - 7.30	0.75	0.17
7.79 - 7.92	0.79	0.14
9.83 - 9.87	0.77	0.05

Figure 27: Screenshot of keyword search in *WinSprofiler* 2.13.

In border control, voice can be easily recorded and processed as a background process for providing additional information and invoking an alert when obvious mismatch appears. In network banking, the person identity should be checked by more reliable means. Since most new laptops already have built-in microphone, speaker verification can be used in the background for providing additional security as soon as voice-based dialogues will be adopted in the software. In these two applications, it would be important to keep the false alarms small enough but being still able to detect certain misuse situations.

In call-center application, realistic cost saving can be achieved assuming that every call answered by human costs roughly 10 times more than when dealt automatically by computer. Voice-based user identification can be used for directing the user faster to the proper service person, or into some extent, automate the service in cases when the customer is seeking for basic information easily found by computer after the person have been identified. Moreover, since the calls happen via somewhat controlled environment (mobile or landline phones), the system is expected to be more reliable than the other applications using remote recordings and unknown conditions.

In helpdesks, personal service is provided but the customer needs to be verified remotely by asking certain questions concerning address, social security number or similar information. Again, voice-based verification could be used if high confidence detection is made during the conversation, or otherwise, it could provide additional security information in case of obvious fraud situations.

In tele-conferencing (Fig. 28), exact identification is not necessarily needed but it might be enough to separate different speakers from each other. Biggest challenges are varying acoustic and technical conditions, and the case of over-lapping speakers.

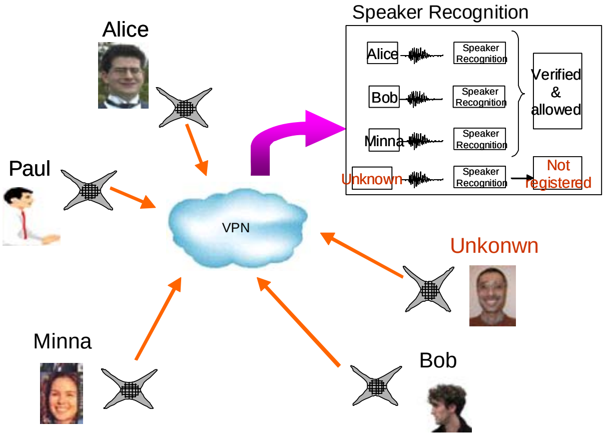
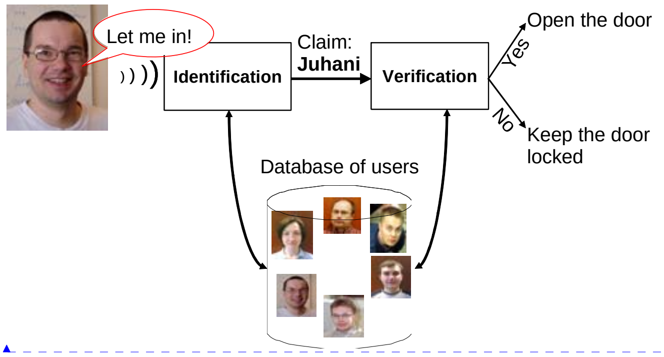


Figure 28: Application scenario for teleconferencing.

In access control (Fig. 29), voice-based system could be used via door phone, or when the access is asked remotely via mobile phone – either as such or combined with password detection. Biggest obstacles to make voice-based system in practice are the need for having at least some level of security without losing convenience that the RFID-based verification has. Uncontrolled voice conditions in distant voice recording in case of door phone and remote voice control gives also technical challenges.

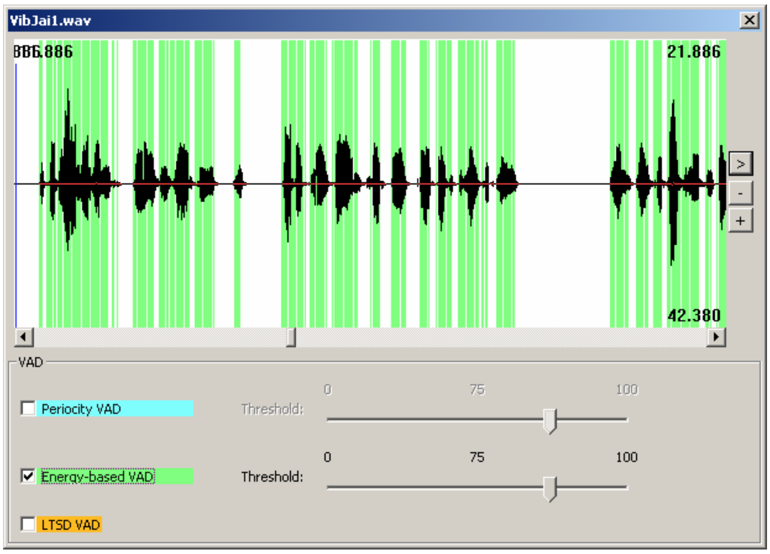


Formatted

Figure 29: Voice-based user authentication in an access control application.

The prototype applications developed during the project have aimed at simulating these potential applications. The main prototypes that have been developed are:

- *WinSprofiler*: Windows-based demonstrator for voice-based identification, large-scale testing of recognition accuracy, and a practical tool for forensic purposes and other similar security applications.
- *VoiceGrep*: Tool for voice detection from long audio recordings (hours or even days). Methods are also implemented in WinSprofiler (Fig. 30).
- *EpocSprofiler*: Mobile phone demonstrators for speaker identification and verification in Symbian operating system, see Fig. 31.
- *DoorSprofiler*: Real door-control system installed in SIPU laboratory in Joensuu Science Park as a demonstrator to simulate access control system, see Fig. 24.



☐ Periocity VAD Threshold:

☑ Energy-based VAD Threshold:

☐ LTSD VAD

Figure 30: Voice activity detection in *WinSprofiler*.

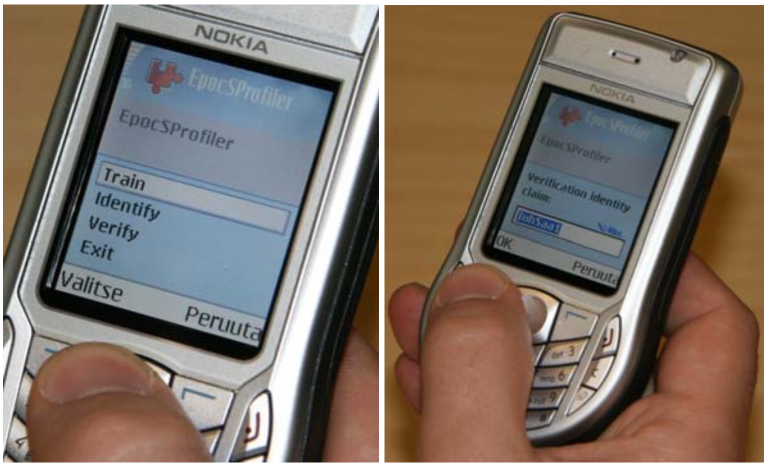


Figure 31: Example of the Symbian demonstrator (EpocSprofiler 2.0)

7. Conclusions and discussion

Main philosophy throughout the project was to study the methods thoroughly to gain deeper understanding about what is essential, how the models should be used, and then implement the settled baseline in practice. This can be seen as a slight lag between the fast developing state-of-the-art, and what has been put into the prototype software as the baseline. However, the gained knowledge makes it possible to transfer new incremental improvements to practical application much faster than mere Matlab prototyping.

Intuitively, longer term features should provide better recognition results. In practice, state-of-the-art techniques have been mostly converged to the use of short term features such as MFCC, LPCC, and their variants (deltas, normalization). Current challenges lay more on the technical side. Normalization due to channel mismatch and change of condition are more vital issues than which features should be used, and our findings confirm this.

The problem of using text-dependent information is a two-sided sword. Intuitively, being able to recognize (and normalize) the unknown speech sample according to its textual content would be useful for speaker recognition. On the other hand, technical matters dominate matching too much, and tuning the speaker models according to the content would not be robust against recognition errors (or assumptions) of the spoken language or exact textual content. Password dependency is also an undesired feature of most recognition systems.

Moreover, current methodology already makes the recognition relative to the content in an indirect (and text-independent) way by using the so-called background model. Thus, any feature matching is already put into a context and only the difference to the assumed background model is what makes the difference in the decision in practice.

As a future work, the following three points are worth to consider:

1. Usability issues in general including: how to setup system fast and easily, and how to train the background model. Can the user models be trained remotely off-site, and can previously recorder training samples used in different conditions.
2. Using the pseudo-phonemes for capturing longer-term text-independent features, similarly as was used for VAD in [Timofte’2007].
3. Study recognition and synthesis as inverse problems. Better understanding how to separate the speech content, acoustic environment and speaker characteristic in the recognition process could be learned better by being able to model speaker identity in synthesis as well. This would require us to be able to build model-based synthesizer based on the same features used in recognition, and to add or remove (short-term) speaker characteristics. If this can be successfully performed, then the key knowledge can be utilized in the opposite, speaker recognition problem as well.

Acknowledgements

The work has been supported by the *National Technology Agency of Finland* (TEKES) as the four year project *New Methods and Applications of Speech Technology* (PUMS) under the contracts 40437/03, 49398/04 40356/05, 40195/06, and by all the government organizations and private companies participating these projects.

The authors also express their warmest thanks to the contributing people in all organizations devoting their time and efforts for the success of the project. The project indeed followed the “from ashes to fame” theme as the Tekes’ program name FENIX implies.

To be continued.

Publications (4 journal, 15 conference, 2 other)

1. T. Kinnunen, M. Tuononen and P. Fränti, "Which clustering algorithm to select for text-independent speaker recognition?", *Pattern Recognition*, 2008. (accepted)
2. V. Hautamäki, T. Kinnunen, I. Kärkkäinen, J. Saastamoinen, M. Tuononen and P. Fränti, "Maximum a posteriori adaptation of the centroid model for speaker verification", *IEEE Signal Processing Letters*, 2008. (accepted)
3. T. Kinnunen, E. Karpov and P. Fränti, "Real-time speaker identification and verification", *IEEE Trans. on Audio, Speech and Language Processing*, 14 (1), 277-288, January 2006.
4. J. Saastamoinen, E. Karpov, V. Hautamäki and P. Fränti, "Accuracy of MFCC based speaker recognition in series 60 device", *Journal of Applied Signal Processing*, (17), 2816-2827, September 2005.
5. V. Hautamäki, M. Tuononen, T. Niemi-Laitinen and P. Fränti, "Improving speaker verification by periodicity based voice activity detection", *Int. Conf. on Speech and Computer (SPECOM’07)*, Moscow, Russia, vol. 2, 645-650, October 2007.
6. T. Kinnunen, E. Chernenko, M. Tuononen, P. Fränti and H. Li, "Voice activity detection using MFCC features and support vector machine", *Int. Conf. on Speech and Computer (SPECOM’07)*, Moscow, Russia, vol. 2, 556-561, October 2007.
7. T. Kinnunen, V. Hautamäki and P. Fränti, "On the use of long-term average spectrum in automatic speaker recognition", Int. Symp. on Chinese Spoken Language Processing (ISCSLP’06), Singapore, Companion volume, 559-567, December 2006.

8. R. Timofte, V. Hautamäki and P. Fränti, "Speaker, vocabulary and context independent word spotting system in continuous speech", Int. Symp. on Chinese Spoken Language Processing (ISCSLP’06), Singapore, Companion volume, 396-407, December 2006.

9. J. Saastamoinen, Z. Fiedler, T. Kinnunen and P. Fränti, "On factors affecting MFCC-based speaker recognition accuracy", *Int. Conf. on Speech and Computer (SPECOM'05)*, Patras, Greece, 503-506, October 2005.

10. H. Gupta, V. Hautamäki, T. Kinnunen and P. Fränti, "Field evaluation of text-dependent speaker recognition in an access control application", *Int. Conf. on Speech and Computer (SPECOM'05)*, Patras, Greece, 551-554, October 2005.

11. T. Kinnunen, R. Gonzalez-Hautamäki, "Long-Term F0 Modeling for Text-Independent Speaker Recognition" *Int. Conf. on Speech and Computer (SPECOM'05)*, Patras, Greece, 567-570, October 2005.

12. T. Niemi-Laitinen, J. Saastamoinen, T. Kinnunen, and P. Fränti, "Applying MFCC-based automatic speaker recognition to GSM and forensic data", *2nd Baltic Conf. on Human Language Technologies* (*HLT’05*), 317-322, Tallinn, Estonia, April 2005.

13. T. Kinnunen, E. Karpov and P. Fränti, "Real-time speaker identification", *Int. Conf. on Spoken Language Processing*, (*ICSLP’04*), Jeju Island, Korea, vol 3, 1805-1808, October 2004.

14. T. Kinnunen, E. Karpov and P. Fränti, "Efficient online cohort selection method for speaker verification", *Int. Conf. on Spoken Language Processing*, (*ICSLP’04*), Jeju Island, Korea, vol 3, 2401-2405, October 2004.

15. J. Saastamoinen, E. Karpov, V. Hautamäki and P. Fränti, "Automatic speaker recognition for series 60 mobile devices", SPECOM'2004, St. Petersburg, Russia, 353-360, September 2004.

16. T. Kinnunen, V. Hautamäki and P. Fränti, "Fusion of spectral feature sets for accurate speaker identification", SPECOM'2004, St. Petersburg, Russia, 361-365, September 2004.

17. E. Karpov, T. Kinnunen and P. Fränti, "Symmetric distortion measure for speaker recognition", SPECOM'2004, St. Petersburg, Russia, 366-370, September 2004.

18. T. Kinnunen, V. Hautamäki and P. Fränti, "On the fusion of dissimilarity-based classifiers for speaker identification", *European Conf. on Speech Communication and Technology, (Eurospeech’2003*), Geneva, Switzerland, 2641-2644, September 2003.

19. T. Kinnunen, E. Karpov and P. Fränti, "A speaker pruning algorithm for real-time speaker identification", *Int. Conf. on Audio- and Video-Based Biometric Person Authentication* (*AVBPA’03*), Guildford, UK, LNCS vol. 2688, 639-646, June 2003.

20. T. Niemi-Laitinen, "Äänet Keskusrikospoliisissa - tutkimusta ja asiantuntijapalvelua." Teoksessa K. Kaartinen (toim.) Rikostorjunnan etulinjassa. Keskusrikospoliisi 80 vuotta, 179-196, 2006.

21. O. Grebenskaya, T. Kinnunen, P. Fränti, "Speaker clustering in speech recognition", *Finnish Signal Processing Symposium (FINSIG'05)*, 46-49, Kuopio, Finland, August 2005.

Theses (1 PhD, 1 PhLic, 6 MSc)

1. Tomi Kinnunen, *Optimizing spectral feature based text-independent speaker recognition*, PhD thesis, University of Joensuu, June 2005.
2. Tomi Kinnunen, *Spectral features for automatic text-independent speaker recognition,* PhLic Thesis, University of Joensuu, Dept. of Computer Science, February 2004.
3. Radu Timofte, *Short-term time series in automatic speech processing*, MSc thesis, Computer Science, University of Joensuu, November 2007.
4. Sergey Pauk, *Use of long-term average spectrum for automatic speaker recognition*, MSc thesis, Computer Science, University of Joensuu, December 2006.
5. Marko Tuononen, *Local search as clustering method*, MSc thesis, Computer Science, University of Joensuu, April 2006. (in Finnish)
6. Olga Grebenskaya, *Speaker clustering in speech recognition*, MSc thesis, Computer Science, University of Joensuu, March 2005.
7. Rosa Gonzalez Hautamäki, *Fundamental frequency estimation and modeling for speaker recognition*, MSc thesis, University of Joensuu, July 2005.
8. Timo Viinikka, *Effect of speech coding to speaker recognition*, MSc thesis, Computer Science, University of Joensuu, December 2004. (in Finnish)

Other references

1. R. Auckenthaler, M. Carey and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems", *Digital Signal Processing*, 10(1-3), pp. 42--54, January 2000.
2. B. Atal, "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification", *Journal of the Acoustic Society of America*, 55(6), pp. 1304--1312, 1974.
3. N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D.A. van Leeuwen, P. Matejka, P. Schwarz, A. Strasheim, "Fusion of Heterogeneous Speaker Recognition Systems in the STBU Submission for the NIST Speaker Recognition Evaluation 2006", *IEEE Trans. Audio, Speech and Language Processing*, 15(7), 2072-2084, Sept. 2007.
4. L. Burget, P. Matejka, P. Schwarz, O. Glembek, J.H. Cernocky, "Analysis of Feature Extraction and Channel Compensation in a GMM Speaker Recognition System", *IEEE Trans. Audio, Speech and Language Processing*, 15(7), 1979-1986, Sept. 2007.
5. W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer and P.A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition", *Computer Speech and Language*, 20(2-3), pp. 210--229, April 2006.
6. S.B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Trans. Acoustics, Speech, and Signal Processing*, 28(4), pp. 357--366, August 1980.
7. ETSI, “Voice Activity Detector (VAD) for Adaptive Multi-Rate (AMR) Speech Traffic Channels”, *ETSI EN 301 708 Recommendation*, 1999.
8. S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification", *IEEE Trans. on Acoustics, Speech and Signal Processing*, 29(2), pp. 254--272, April 1981.
9. H. Hermansky and N. Morgan, "RASTA processing of speech", *IEEE Trans. Speech & Audio Processing*, 2(4), pp. 578--589, October 1994.
10. ITU, “A Silence Compression Scheme for G.729 Optimized for Terminals Conforming to Recommendation V.70”, *ITU-T Recommendation G.729-Annex B*, 1996.

11. S.M. Kay, *Fundamentals of Statistical Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1998.

12. N. Kumar and A.G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition", *Speech Communication*, vol. 26, pp. 283--297, 1998

13. B. Ma, D. Zhu, R. Tong, H. Li, "Speaker Cluster based GMM tokenization for speaker recognition", *Proc. Interspeech* 2006, pp. 505--508, Pittsburg, USA, September 2006.

14. B. Ma, H. Li, R. Tong, "Spoken Language Recognition Using Ensemble Classifiers" *IEEE Trans. Audio, Speech and Language Processing*, 15(7), 2053-2062, Sept. 2007.

15. J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification", *Proc. Speaker Odyssey* 2001, pp. 213--218, June 2001, Crete, Greece.

16. J. Ramirez, J.C. Segura, C. Benitez, A. de la Torre and A. Rubio, “Efficient voice activity detection algorithms using long-term speech information”, *Speech Communications*, vol. 42(3-4), pp. 271--287, 2004.

17. D.A. Reynolds and T.F. Quatieri and R.B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, 10(1), pp. 19--41, January 2000.

18. Speaker and Language Characterization Interest Group, *Speaker Odyssey*, Puerto Rico, June 2006. http://www.speakerodyssey.com/odyssey_2006.htm

19. M. Turunen, J. Hakulinen, K.-J. Räihä, E.-P. Salonen, A. Kainulainen, and P. Prusi, “An architecture and applications for speech-based accessibility systems,” *IBM Systems Journal*, vol. 44, pp. 485-504, 2005.

20. R. Tong, B. Ma, K.A. Lee, C. You, D. Zhou, T. Kinnunen, H. Sun, M. Dong, E.S. Chng, H. Li, "The IIR NIST 2006 Speaker Recognition System: Fusion of Acoustic and Tokenization Features", *Int. Symp. on Chinese Spoken Language Processing (ISCSLP’06)*, Singapore, 566--577, December 2006.