VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

QUERY-BY-EXAMPLE KEYWORD SPOTTING

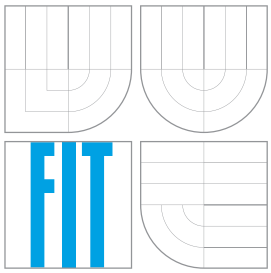
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MIROSLAV SKÁCEL

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE KLÍČOVÝCH SLOV ZADANÝCH VZOREM

QUERY-BY-EXAMPLE KEYWORD SPOTTING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MIROSLAV SKÁCEL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. IGOR SZÖKE, Ph.D.

BRNO 2015

Master Thesis Specification/17862/2014/xskace00

Brno University of Technology - Faculty of Information Technology

Department of Computer Graphics and Multimedia Academic year 2014/2015

Master Thesis Specification

For: **Skácel Miroslav, Bc.**

Branch of study: Computer Graphics and Multimedia

Title: **Query-by-Example Keyword Spotting**

Category: Speech and Natural Language Processing

Instructions for project work:

1. Get acquainted with the theory of keyword spotting (KWS) and keyword spotting where the query is provided as an audio example (QbE).
2. Choose a development and evaluation data setup. Choose a scoring metric.
3. Study state-of-the-art approaches for QbE and try to reimplement, evaluate, and analyse them.
4. Suggest and implement new techniques to improve the QbE.
5. Conclude and discuss your findings.
6. Create an A2 poster and a short video presenting your work.

Basic references:

- Follow advices of the supervisor.

Requirements for the semestral defense:

Items 1-3 of the assigment.

Detailed formal specifications can be found at http://www.fit.vutbr.cz/info/szz/

The Master Thesis must define its purpose, describe a current state of the art, introduce the theoretical and technical background relevant to the problems solved, and specify what parts have been used from earlier projects or have been taken over from other sources.

Each student will hand-in printed as well as electronic versions of the technical report, an electronic version of the complete program documentation, program source files, and a functional hardware prototype sample if desired. The information in electronic form will be stored on a standard non-rewritable medium (CD-R, DVD-R, etc.) in formats common at the FIT. In order to allow regular handling, the medium will be securely attached to the printed report.

Supervisor: **Szőke Igor, Ing., Ph.D.**, DCGM FIT BUT

Beginning of work: November 1, 2014

Date of delivery: May 27, 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

Jan Černocký
Associate Professor and Head of Department

Abstrakt

Tato diplomová práce se zabývá moderními přístupy detekce klíčových slov a detekce frází v řečových datech. V úvodní části je seznámení s problematikou a teoretický popis metod pro detekci. Následuje popis reprezentace vstupních datových sad použitých při experimentech a evaluaci. Dále jsou uvedeny metody pro detekci klíčových slov definovaných vzorem. Následně jsou popsány evaluační metody a techniky použité pro skórování. Po provedení experimentů na datových sadách a po evaluaci jsou diskutovány výsledky. V dalším kroku jsou navrženy a poté implementovány moderní postupy vedoucí k vylepšení systému pro detekci a opět je provedena evaluace a diskuze dosažených výsledků. V závěrečné části je práce zhodnocena a jsou zde navrženy další směry vývoje našeho systému. Příloha obsahuje manuál pro používání implementovaných skriptů.

Abstract

The aim of the thesis is to get acquainted with modern approach of keyword spotting and spoken term detection in speech data. The bases of keyword spotting are described at first. The data representation used for experiments and evaluation are introduced. Keyword spotting methods where query is provided as an audio example (Query-by-Example) are presented. The scoring metrics are described and experiments follow. The results are discussed. Further, modern approaches of keyword spotting are suggested and implemented. The system with new techniques is evaluated and the discussion of results achieved follows. The conclusions are drawn and the discussion of future directions of development is held. The Appendix contains user manual for using implemented system.

Klíčová slova

dotaz vzorem, detekce klíčových slov, DTW, dynamické borcení času, STD, detekce frází v řeči, dynamické programování

Keywords

Query-by-Example, keyword spotting, DTW, Dynamic Time Warping, STD, Spoken Term Detection, dynamic programming

Citace

Miroslav Skácel: Query-by-Example Keyword Spotting, diplomová práce, Brno, FIT VUT v Brně, 2015

Query-by-Example Keyword Spotting

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Igora Szökeho, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal. Některé prezentované části systému byly vypracovány Igorem Szökem nebo Lukášem Burgetem a jsou v této práci zahrnuty z důvodu pochopení kompletního systému. Tyto části jsou vždy označeny a pro detailnější informace odkázány na literární zdroje.

.....
Miroslav Skácel
May 28, 2015

Poděkování

Chtěl bych poděkovat vedoucímu své diplomové práce panu Ing. Igoru Szökemu, Ph.D., za poskytnuté odborné rady, nástroje a řečová data pro experimenty při tvorbě této práce. Dále bych rád poděkoval panu Ing. Lukáši Burgetovi, Ph.D., a panu Ing. Michalu Fapšovi, Ph.D., za cenné rady, nápady a užitečné nástroje a panu Ing. Kamilu Chalupníčkovi za technickou podporu. Rád bych také poděkoval vedoucímu výzkumné skupiny Speech@FIT panu doc. Dr. Ing. Janu Černockému za podporu a trpělivost během tvorby této práce a rovněž i všem ostatním členům této skupiny.

Contents

Chapter 1

Introduction

The amount of speech data constantly increases every year. These data are either processed in real time or stored for further processing. Devices utilising a speech processing are for example cell phones, personal computers and nowadays on-board car navigations and controls, security systems or intelligent houses. Therefore, keyword spotting has became an important topic in speech processing field. Words of interests, such as voice commands or suspect words in secure areas, are detected. It is challenging to investigate modern approaches of methods for detecting words in speech.

The purpose of this thesis is to focus on audio and speech data with low-resource conditions. The problem is defined as finding of spoken words or word phrases (or music samples as well) in other speech (audio) data without any additional knowledge of the target language. The sound quality of speech can be variable. Our task is to implement system or improve an existing one to search for user specified speech cut samples. The system should detect the segments in speech data similar to user specified speech cuts and return useful information of these found detections. The implemented system should be based on known algorithms for spoken term detection and further improvement adopted from the latest approaches should be done. The performance of the system should be represented in a form comparable to other systems dealing with the same task.

The implemented system will participate into spoken term detection task which is held by speech processing community every year. The competition among all participants will run. The goal is to search for hundreds of defined speech samples into thousands of speech utterances. Different conditions to search speech samples are set. All systems will be scored by a newly introduced scoring metric.

The thesis is structured into several essential parts. The first step will be a reflection of input data representation. We will need to extract useful features from the raw speech data. Further, a deep analysis of methods for described problem will follow. The aim will be to exploit techniques for detecting similar segments in speech data using dynamic alignment of speech samples. We will compare speech data to each other with different metrics. Further, we will reimplement an algorithm suitable to deal with the given issue. The different data sets will be exploited for experiments with the system. We will focus on the accuracy of the system compared to other systems. At last, the improvement of the system will be made and the comparison will follow.

Chapter 2

Bases of keyword spotting and Query-by-Example

This chapter provides description of spoken term detection and keyword spotting methods for an audio data, where query is provided as an audio example, called Query-by-Example. The utilisation of these techniques on speech data is outlined. The decomposition of the spoken term detection system and the description of essential parts follows. Further, the different input data representations for the system are detailed. The data sets used in performed experiments are presented. The scoring methods and evaluation metrics exploited to score results of developed system are described. The related work containing the adopted ideas and algorithms for further improvement of the system is presented. The theory of spoken term detection, keyword spotting, Query-by-Example and scoring metrics is adopted from [23][19][4].

2.1 Spoken Term Detection and keyword spotting

Spoken Term Detection (STD) is a technique to find a list of terms fast and accurately in audio data. Terms are meant as single words or sequences of words and are represented in a textual form. This method is often denoted as textual STD. It is assumed to have enough text resources and knowledge of the target language to exploit this method. STD systems are usually built and dependent on speech recognizers. This is the reason why textual STD methods are not suitable for term detection with low resources. On the other hand, a demand for development of STD systems for low resource languages or completely missing resources (e.g. security field) rises. If it is not possible to train the target language-specific acoustic models then the system needs to be trained in the language-independent way. Especially in the cases where a user has no knowledge of the textual representation of the term to search or it is required to enter the term as a voice command. Therefore, the Query-by-Example STD technique was proposed [23]. In Figure 2.2, our term detection system is depicted.

Keyword spotting (KWS) is a similar method to STD. The difference is that keywords consist of a single isolated word. If a phrase containing several words occurs on the input, it is still taken as a single object. These systems are usually based on speech recognizers (e.g. Acoustic KWS, more in [19]) but that topic is beyond the scope of this thesis. We are interested into KWS systems based on other systems that do not understand speech [19].

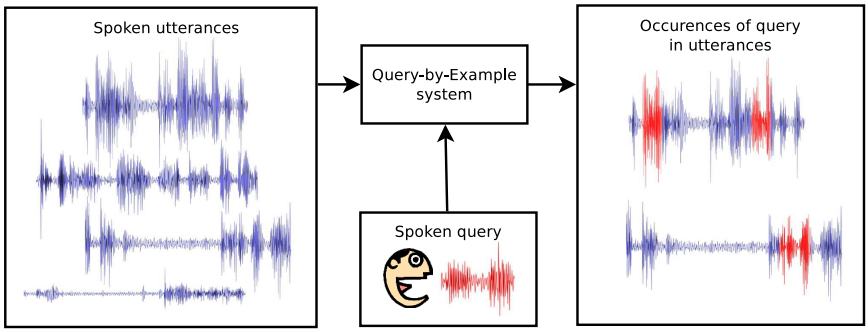


Figure 2.1: *Query-by-Example system. User-defined spoken query is searched in a database of spoken utterances, providing the user with occurrences of the query. The query can be defined by direct input from a microphone or by a region of speech selected in another utterance.* [4].

2.2 Query-by-Example

Query-by-Example (QbE) is a method to search an example of an object or at least a part of it in some other objects. QbE has been used mostly in applications like sound classification, music retrieval or spoken document retrieval. The example of an object to find is called query and in our task, it consists of the spoken term to search. The spoken term is a word or a word phrase and it is represented as a speech (or music) cut. The user can specify one or more cut instances containing the term of interest. This query is then searched in data pool (e.g. set of speech utterances) and segments that are similar to the searched query are returned. The method relies only on a spoken term example as an input on the contrary to a textual input for textual STD. Therefore, it is called Query-by-Example Spoken Term Detection (QbE STD). QbE STD is used when not enough resources for training acoustic models are available and so it is impossible to use large vocabulary continuous speech recognition (LVCSR) system to conduct textual STD. Hence, usage of LVCSR is impossible for a language-independent QbE STD [23]. There are three main approaches to build QbE STD systems on: a template matching, a sequential statistical modelling and a lattice matching. We are focused on the first approach where we compare input features between themselves. This approach exploit the dynamic programming technique called Dynamic Time Warping. This method is used to compare speech patterns and confront inconsistencies in time (more details in Chapter 3).

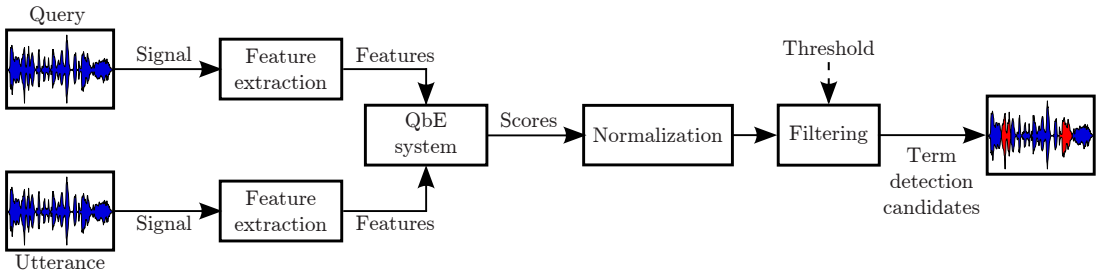


Figure 2.2: *Block diagram of spoken term detection system based on Query-by-Example* [19].

A scheme of our spoken term detection system is shown in Figure 2.2. The whole process of a detection can be separated into several phases. A raw signal enters the system inputs. The extractors generate vectors of speech features. The features are then processed by Query-by-Example system. The outputs of QbE are records containing a query-utterance pair, a starting and ending time of the detection and a confidence score. The score is normalized (e.g. by term length or according to scoring metric). The threshold is applied to the scores to filter out bad detection and reduce false alarms which is important for the scoring.

2.3 Feature extraction

An application of raw audio data as an input to our implemented algorithm would be very inefficient so we used *phnrec*[1] [16] tool developed at Brno University of Technology to extract speech features. The extractor is based on Hidden Markov Model (HMM)/Artificial Neural Network (ANN) hybrids and was trained on TIMIT[2], SpeechDat-E [10] (SD) and GlobalPhone (GP) corpora. We used phoneme-state posteriors (POST) and bottlenecks (BN).

2.3.1 Phoneme-state posteriors

The input is raw audio data with speech. Speech is segmented into 25 ms long frames. Mel filter banks energies are calculated. For each band, the temporal evolution of energy is taken and vectors are split into right and left parts (therefore the system is called LC-RC - Left Context/Right Context). Each part is windowed by corresponding half of Hamming window. The linear DCT transformation is performed to decorrelate and reduce dimensions. The next step is neural networks trained to estimate probability of phonemes for each of vectors. The concatenation, the transformation and the normalization of vectors follows. The last step comprises the Viterbi decoder to decode the phoneme posteriors [16][17]. Each step of the described system is depicted in Figure 2.3.

Features	# dims
SD CZ POST	138
SD HU POST	186
SD RU POST	159
GP CZ POST	120
GP EN POST	120
GP GE POST	126
GP PO POST	102
GP RU POST	156
GP SP POST	102
GP TU POST	90
GP VI POST	102

Table 2.1: *List of phoneme-state posterior features and the number of their dimensions depending mainly on the number of phonemes for the given language. SD stands for features extracted using SpeechDat-E database, GP using GlobalPhone database.*

[1] http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context
[2] https://catalog.ldc.upenn.edu/LDC93S1

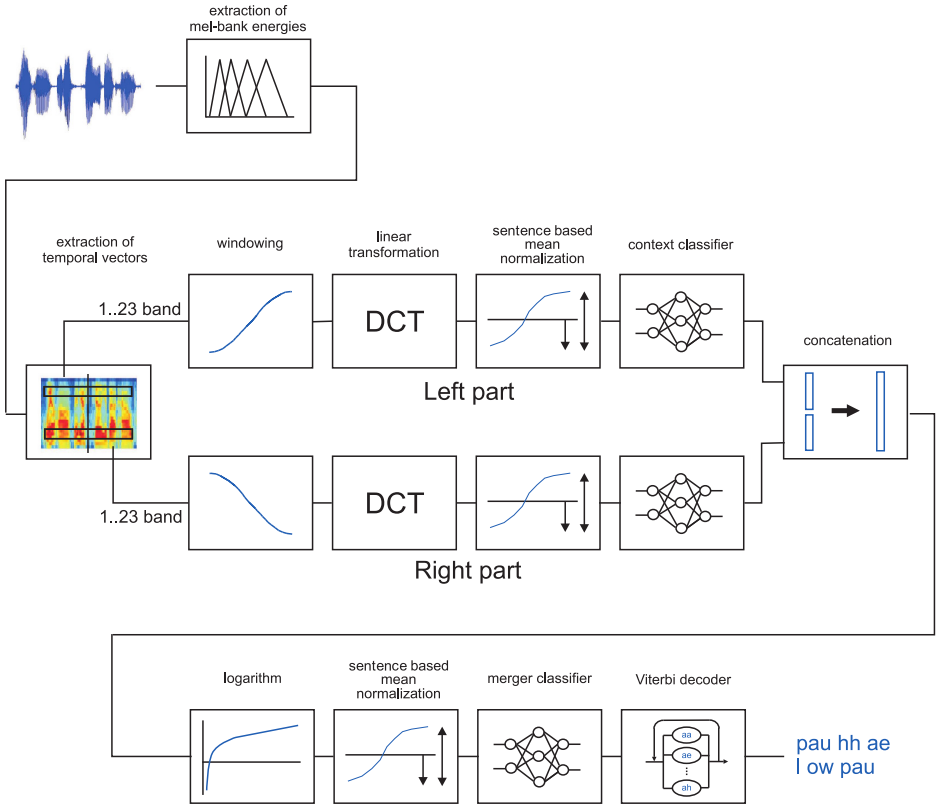


Figure 2.3: *Block diagram of the Split Temporal Context system* [16].

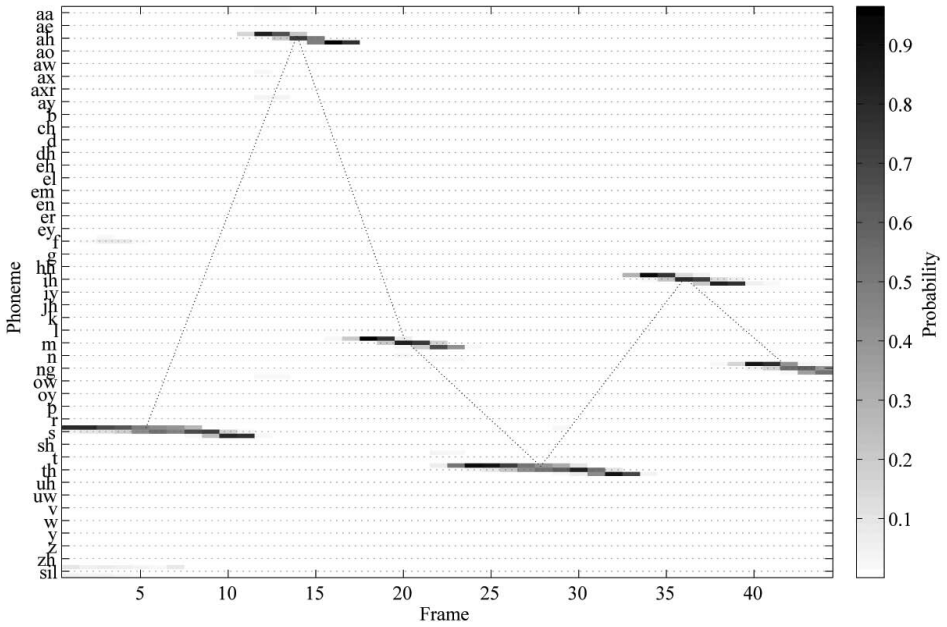


Figure 2.4: *Example of 3-state phoneme posterior probabilities for word* "something". *Each frame states the probability in the range from 0.0 to 1.0 for given phonemes. The word something was estimated as the sequence of phonemes s - ah - m - th - ih - ng* [17].

The final representation of phoneme state features is a sequence of N-dimensional vectors containing phoneme-state posterior probabilities. We generated *3-state phoneme posteriors* (POST) based on different databases and languages. The vectors dimensionality for each database and language is listed in Table 2.1. In Figure 2.4, an example of 3-state phoneme posterior is depicted.

2.3.2 Bottlenecks

The *bottleneck* (BN) features were extracted by hierarchical neural network. BNs are linear outputs (compressed information) of neurons in bottleneck area of ANN topology. It was proved that BN features represents the underlying information better than the probabilistic features.

As for phoneme-state posteriors, speech is segmented into frames. Fourier spectrum is calculated for each frame. Band limited triangular functions called Mel filter banks are applied to get energies. The logarithm of the energies is calculated which corresponds to a human ear perception. The sentence mean normalization follows. Five consecutive frames are used to add information about a temporal evolution. Hamming window is applied followed by linear DCT transformation. The outputs of contextual ANN are stacked on each other and this is taken as an input for the second (merging) ANN for every fifth frame. The size of BN layer in the second ANN is 30 neurons [24][5]. Hence, all BN features used in this thesis are 30 dimensional vectors. In Figure 2.5, the whole process is depicted.

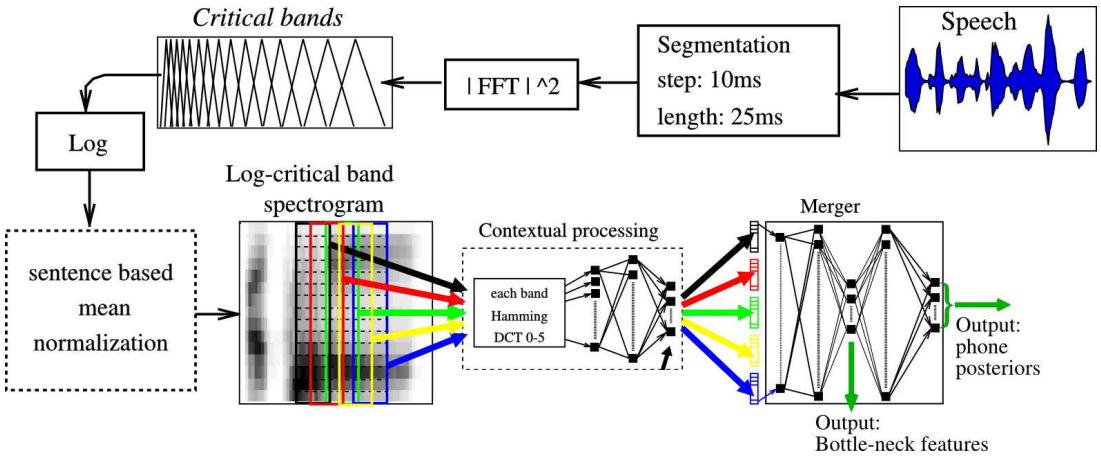


Figure 2.5: *Block diagram of Bottleneck features extraction* [5].

2.4 Evaluation and development datasets

We define four different data sets to test and evaluate implemented spoken term detection system. The first data set is GlobalPhone corpus developed with collaboration of Karlsruhe Institute of Technology (KIT) to provide real read speech data. The second data set was used in MediaEval benchmark in Spoken Web Search (SWS) task in 2012. The third data set was used in the same benchmark in SWS task one year later in 2013. The last data set was used at MediaEval benchmark where SWS task was renamed to Query-by-Example Search on Speech Task in 2014.

2.4.1 GlobalPhone

GlobalPhone (GP) database is a multilingual speech and text database developed at Karlsruhe University at Institute of Technology. This database contains high-quality read speech in a large variety of languages which is suitable for the development of speech recognition systems in many languages. GP consists of 20 languages[3] and was designed to be uniform across languages with respect to the amount of data and speech quality. The read text for each language was selected from local newspapers and was read by about 100 speakers. The text was read by both genders with a variety of age. The speech was microphone recorded with the same conditions for all languages and spontaneous effects like stuttering, false starts, breathing, hesitation and laughing are included [14][15].

languages	CZ	EN	GE	PO	RU	SP	TU	VI
audio format	WAV							
sampling rate	8 kHz							
bit depth	16 bit							
channel	mono, linear							
queries	59	26	36	42	72	81	66	66
dev. data	656	503	1071	481	868	510	664	1161
test. data	687	546	804	480	1179	472	627	1404
train. data	10994	7138	7959	6854	8877	4713	5319	16270

Table 2.2: *Summary of used languages from GlobalPhone database.*

This database was used to train bottlenecks decoders. Only 8 of 20 languages contained in this database was used: namely it is Czech, English, German, Portuguese, Russian, Spanish, Turkish and Vietnamese language. The queries and audio content are separated for each language. The summary for languages used for evaluation from GP database is in Table 2.2.

2.4.2 Spoken Web Search 2012

The *Spoken Web Search*[4] (SWS) task is held every year at MediaEval workshop. The task involves searching for audio content within audio content using an audio query. This task is interesting especially for speech researchers in area of spoken term detection and speech processing with low-resource audio. The task requires to build language-independent audio search system.

The 2012 data set consists of two languages. Each language is divided into two parts: the first is the set of audio queries, the second is set of audio content. Both sets are separated to development and evaluation subsets, which are from the same language. Some of the queries overlap partially. Audio ground through files were generated following the format defined by NIST STD in 2006.

The two represented languages are Indian and African. Let us focus on African language since Indian one is not used for our evaluation. The African audio data were extracted from Lwazi ASR corpus and contains speech from four of eleven different African languages

[3] Arabic, Bulgarian, Chinese, Croatian, Czech, French, German, Hausa, Japanese, Korean, Portuguese, Polish, Russian, Spanish, Swedish, Tamil, Thai, Turkish, Ukrainian, Vietnamese

[4] http://multimediaeval.org/mediaeval2012/sws2012/

language	African
audio format	WAV
sampling rate	8 kHz
bit depth	16 bit
channel	mono, linear
dev. queries	100 (25 per language)
dev. data	1580 (395 per language)
eval. queries	100
eval. data	1660

Table 2.3: *Spoken Web Search 2012 data set for African language.*

(isiNdebele, Siswati, Tshivenda, Xitsonga). Audio consists of a combination of read and elicited speech collected over a telephone channel. Audio recording artifacts can be found in the data [7]. The language summary is in Table 2.3.

2.4.3 Spoken Web Search 2013

The *Spoken Web Search* 2013[5] task was held in the similar way. The difference is in the data sets. The data were expanded on MediaEval 2011 and 2012 SWS tasks by increasing the size of data sets and the number of languages (non-native English, Albanian, Czech, Basque, Romanian and four African). These languages were recorded in different acoustic conditions. The data set is composed of 20 hours of speech and is over 5 times the size of 2012 database [1].

The 2013 data set consists of two parts again: set of queries content and set of audio content. The development and evaluation data has each own query content set but audio content is the same for both. A basic sets of queries consist of about 500 files each and in addition, for some of the queries there are alternative spoken instances to be used in extended runs. Both the queries and audio content were scrambled and randomized. The summary of SWS 2013 dataset is in Table 3.1.

languages	9 (combined)
audio format	WAV
sampling rate	8 kHz
bit depth	16 bit
channel	mono, linear
dev. queries	505 (1551 including ext. queries)
dev. data	10762 (same data as for eval)
eval. queries	503 (1540 including ext. queries)
eval. data	10762 (same data as for dev)

Table 2.4: *Spoken Web Search 2013 data set summary.*

[5] http://multimediaeval.org/mediaeval2013/sws2013/

2.4.4 Query-by-Example Search on Speech Task 2014

The *Query-by-Example Search on Speech Task*[6] (QUESST) data set consists of speech data were collected at several institutions. The corpus is composed of 23 hours of speech in 6 languages (Albanian, Basque, Czech, non-native English, Romanian, Slovak) with various number if audio per language. The search utterances were automatically extracted from longer recordings and checked manually for unwanted qualities. The queries to be searched were recorded manually to avoid previous problems developed from cutting queries from utterances. Speakers maintained a normal speech and a clear speaking style. All data have PCM encoding at 8kHz, 16bits/sample and WAV format. The database has one set of utterances for both development and evaluation. The queries are split into two sets for each part of the task [2]. The summary of database can be seen in Table 2.5.

languages	6
audio format	WAV
sampling rate	8 kHz
bit depth	16 bit
channel	mono, linear
dev. queries	560
dev. data	12 492
eval. queries	555
eval. data	12 492

Table 2.5: *Query-by-Example Search on Speech Task 2014 data set summary.*

Unlike the other presented data sets, this database contains three different types of queries denoted as Type 1, Type 2 and Type 3. *Type 1* of query consists of a spoken term that matches exactly a term in an utterance. *Type 2* of query is a query with variant matching. The query can slightly differ either at the beginning or at the end of the match. The minimum length of a query to match was set to 250 ms and non-matching part was required to be smaller than the matching part. As an example, the query containing the term „*researcher*“ would match the term „*research*“ or „*searcher*“ in an utterance. *Type 3* of query contains a phrase of several terms. To consider a query match, all terms in the phrase have to match but not necessarily in the same order as stored in a query or there can be a filler (a silence or extraneous terms) between terms of the phrase. For example, the phrase „*curious researcher*“ would match an utterance with the phrase „*this researcher is really curious*“ or „*the curiousest research*“. Note that there are no silences or marks between the terms of a phrase in a query Type 3 [2].

2.5 Scoring metrics

The proposed system was evaluated for its accuracy. There are two different approaches to evaluate the performance of the system. The first one evaluates the position and the score of the detection while the other one takes into account only the score regardless the position of the detection. The score represents a confidence. The main aspect of the system performance is a type and a quality of an input source.

[6] http://multimediaeval.org/mediaeval2014/quesst2014/

As one of measurements for our evaluation, we used the metrics defined by NIST STD[7] in 2006. Each query match detection has a start time, a end time and a confidence which is a value a score) saying sureness of the spoken term detector about the detection of the query. All detections were scored in the same manner, the higher value of confidence the higher probability of correct term detection. The detections are compared with a reference transcription and marked as a hit, a miss or a false alarm (see Figure 2.6). The good system has maximum number of hits and minimum number of FAs and misses.

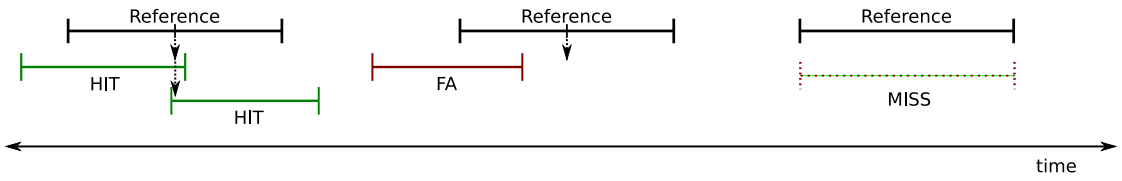


Figure 2.6: *Examples of HIT, FA (False Alarm) and MISS* [19].

2.5.1 Term Weighted Value

Term-weighted Value (TWV) was defined as the primary metric for NIST STD 2006 evaluations and was used to measure overall system detection performance. TWV is scalar metric designed for comparison of different spoken term detection systems. It assigns positive value for every correct output and negative value for every incorrect output. The requirement a query to be called a hit is relaxed within 0.5 s range from the reference time span. If other overlapping detections occur, they are considered as false alarms so only one detection to one reference counts. An interesting fact about TWV is that a miss is much more expensive (meant that term score is worse) for less occurring terms than for more frequently occurred terms but contrary, false alarm is equally expensive for both more or less occurring terms. [19].

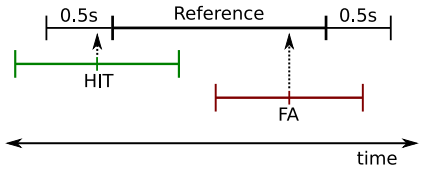


Figure 2.7: *Examples of HIT and reference overlap defined by NIST for STD evaluation and TWV metric. If two or more detections overlap one reference, only one is considered as HIT and the other is considered as FA* [19].

Miss and false alarm probabilities are calculated for each query, the query specific value over all queries are computed and then, by averaging these values, an overall system score is obtained.

The miss error rate (probability of miss) p_{miss} for a query q and a threshold θ is defined by [11]:

$$p_{miss}(q,\theta) = \frac{N_{miss}(q,\theta)}{N_{act}(q)}, \tag{2.1}$$

[7] http://www.itl.nist.gov/iad/mig/tests/std/

$$p_{miss}(\theta) = \underset{q}{average}\{p_{miss}(q,\theta)\}, \tag{2.2}$$

where $N_{miss}(q,\theta)$ is the number of miss errors corresponding to query q and threshold θ, $N_{act}(q)$ is the amount of actual occurrences of query q.

The false alarm error rate (probability of false alarm) p_{fa} for query q and threshold θ is defined as [11]:

$$p_{fa}(q,\theta) = \frac{N_{fa}(q,\theta)}{N_{nt}(q)}, \tag{2.3}$$

$$p_{fa}(\theta) = \underset{q}{average}\{p_{fa}(q,\theta)\}, \tag{2.4}$$

where $N_{fa}(q,\theta)$ is the amount of false alarm errors corresponding to query q and threshold θ, $N_{nt}(q)$ is the number of non-target trials. Finally, TWV is defined by [11]:

$$TWV(\theta) = 1 - (p_{miss}(\theta) + \beta \cdot p_{fa}(\theta)), \tag{2.5}$$

where β is a weight factor defined by [11]:

$$\beta = \frac{C_{fa} \cdot (1 - p_{target})}{C_{miss} \cdot p_{target}}, \tag{2.6}$$

where $C_{miss} > 0$ is the cost of miss and $C_{fa} > 0$ is the cost of false alarm, $p_{target} \in [0,1]$ is the prior probability of a target trial. $TWV(\theta)$ range falls into the interval $[-\beta, 1]$ where 1 stands for a perfect system, 0 for a system rejecting all the trials and $-\beta$ for the worst possible system.

2.5.2 Actual TWV

Actual TWV (ATWV) is calculated by hard decision for each detection given by a system. ATWV can be an extremely unstable performance measure [4]. The best system score is ATWV = 1. Lower value of ATWV means worse accuracy of the system. Note that a score can even be a negative number [22][2]. Formally [11]:

$$ATWV = TWV(\theta_{act}), \tag{2.7}$$

where θ_{act} is a specified hard threshold.

2.5.3 Maximum TWV

Maximum TWV (MTWV) defines the global upper bound for ATWV, formally [4]:

$$MTWV = TWV(\theta_{opt}), \tag{2.8}$$

where θ_{opt} is the global optimal threshold for all queries:

$$\theta_{opt} = \arg\max_{\theta}\{TWV(\theta)\} \tag{2.9}$$

2.5.4 Upper Bound TWV

Upper Bound TWV (UBTWV) has individual threshold for each query. The ideal term threshold value is found to get maximum TWV for each term by the following equation [22]:

$$UBTWV = 1 - \underset{q}{average}\{p_{miss}(q, \theta_{opt}(q)) + \beta \cdot p_{fa}(q, \theta_{opt}(q))\}, \tag{2.10}$$

where $\theta_{opt}(q)$ is the optimal threshold for each query:

$$\theta_{opt}(q) = \arg\max_{\theta}\{TWV(q, \theta)\} \tag{2.11}$$

2.5.5 Normalized cross entropy C_{nxe}

Normalized cross entropy cost (C_{nxe}) measures the fraction of information, with regard to the ground truth, that is not provided by system scores, assuming that they can be interpreted as log-likelihood ratios. The best system score is $C_{nxe} \approx 0$ and a non-informative (random) system returns $C_{nxe} = 1$. System scores $C_{nxe} > 1$ indicate severe miscalibration of the log-likelihood ratio scores. C_{nxe} is computed on system scores for a reduced subset of all the possible set of trials. Each trial consists of a query q and a segment x. For each trial, the ground truth is a $True$ or $False$ depending on whether q actually appears in x or not [2][18].

More formally, the empirical cross entropy [11]:

$$C_{xe} = \frac{1}{\log 2}\left(\frac{p_{target}}{|T_{true}(\mathcal{S})|}\sum_{t\in T_{true}(\mathcal{S})} C_{\log}(llr_t) + \frac{1-p_{target}}{|T_{false}(\mathcal{S})|}\sum_{t\in T_{false}(\mathcal{S})} C_{\log}(llr_t)\right), \tag{2.12}$$

where $T_{true}(\mathcal{S})$ is the set of target trials, $T_{false}(\mathcal{S})$ is the set of non-target trials, $C_{\log}(llr_t)$ the logarithmic cost function.

The empirical cross entropy called the prior entropy is defined by [11]:

$$C_{xe}^{prior} = \frac{1}{\log 2}\left(p_{target}\cdot\log\frac{1}{p_{target}} + (1-p_{target})\cdot\log\frac{1}{1-p_{target}}\right) \tag{2.13}$$

Last, the normalized cross entropy defined by [11]:

$$C_{nxe} = \frac{C_{xe}}{C_{xe}^{prior}} \tag{2.14}$$

2.5.6 Minumum C_{nxe}

The cross entropy measures both the discrimination between target and non-target trial and the calibration. To estimate the calibration loss, a system can be optimally recalibrated using a simple reversible transformation, such as [11]:

$$\hat{llr} = \gamma \cdot llr + \delta, \tag{2.15}$$

where llr are log-likelihood ratios, γ and δ are calibration parameters that can be used to minimize the normalized cross entropy [11]:

$$C_{nxe}^{min} = \min_{\gamma,\delta}\{C_{nxe}\}, \tag{2.16}$$

and the calibration loss is $C_{nxe} - C_{nxe}^{min}$.

2.6 Related work

We provide a survey of papers and other literary sources the ideas for the system improvement have been taken from. Procedures, methods or algorithms, that are interesting for our work, are pointed out. Following sources are considered as the related work:

Meinard Müller: Information Retrieval for Music as Motion (2007) [8]. In this book, the author describes concepts and algorithms for robust and efficient information retrieval using two different types of multimedia: waveform-based music data and human motion data. Several approaches in music information retrieval are discussed. The author focuses on efficient strategies for music synchronization, audio matching, audio structure analysis, motion analysis, retrieval and classification. We studied Chapter 4 of this book where the well-known method called Dynamic Time Warping (DTW) is described in detail. This dynamic programming method is fundamental of our implemented algorithm.

Javier Tejedor et al.: Comparison of Methods for Language-dependent and Language-independent Query-by-Example Spoken Term Detection (2012) [23]. In this article, the Query-by-Example (QbE) Spoken Term Detection (STD) is investigated. The query is entered as speech data or is spoken by a user. Two different features are used for experiments: the phoneme-state posteriors and the bottlenecks. Three QbE systems are described: the first one is based on Gaussian Mixture Model/Hidden Markov Model, the second is based on DTW and the last on Weighted Finite-State Transducers. The results are shown on four different languages. The evaluation shows that DTW system performs the best with a language-dependent setup whereas GMM/HMM works the best with a language-independent setup which is interesting for cases with a lack of standard resources to build ASR system.

Igor Szöke et al.: BUT SWS 2013 - Massive Parallel Approach (2013) [20]. This paper describes QbE system composed of a set of subsystems (called atomic systems) where a half of them is based on Acoustic Keyword Spotting (AKWS) and another half on DTW. The system is using the phoneme-state posterior features. The unsupervised adaptation of the artificial neural network is performed on the target data and features are regenerated then. Voice Activity Detection (VAD) is applied on these features which rapidly increases the system accuracy. The system results are calibrated by mode normalization (m-norm) to deal with different score distributions. The single atomic system exploiting DTW was fundamental for our implemented algorithm and its modification is referred as the baseline system in this thesis.

Luis J. Rodriguez-Fuentes et al.: High-performance Query-by-Example Spoken Term Detection on the SWS 2013 Evaluation (2014) [12]. In this paper, QbE system using an iterative DTW with heuristic pruning is presented. The system achieved the best performance and was the winning one in Spoken Web Search (SWS) task in MediaEval 2013. The phoneme-state posteriors are used as input features and a distance matrix normalization follows. VAD is performed by discarding speech frames where non-speech posterior has the highest value. The score is calibrated exploiting a zero-mean and a unit-variance for each query followed by a majority voting. The results show that the usage of multiple examples per query improves the performance of the system. In this thesis, we experimented with Voice Activity Detection, the distance matrix normalization, the sum-

ming of phoneme-states and a concatenation of input features.

Haipeng Wang et al.: Using Parallel Tokenizers with DTW Matrix Combination for Low-resource Spoken Term Detection (2013) [26]. This paper presents QbE system exploiting parallel subsystems (tokenizers) where each subsystem extracts features from raw speech and calculates a distance matrix for input pair query-utterance then. Those matrices are derived into a combined distance matrix. DTW is applied to this combined matrix. Besides phoneme-state posteriors, GMM and Acoustic Segment Model (ASR) are used as input features. The score normalization is performed. The experiments show that combining parallel subsystems with different tokenizers outperformed the best single subsystem and a derived distance matrix works better when more than 3 subsystems are involved. We used the idea of parallel subsystems generating a distance matrix and a combination of them described in this paper.

Haipeng Wang and Tan Lee: The CUHK Spoken Web Search System for MediaEval 2013 (2013) [25]. In this paper, an improvement of the QbE system from previously mentioned paper are described. The different subsystems based on Gaussian Component Clustering (GCC) are presented. The second update is a query expansion based on the technique called Pitch Synchronous OverLap and Add (PSOLA). For our work, an interesting update is a score normalization employing scaling, exponential function, mean and variance normalization.

2.7 Conclusion

We defined the task of this thesis as searching of a user-defined spoken query in a database of spoken utterances. The approaches for detection of a spoken term/keyword based on QbE were presented. Our baseline QbE STD system we will use for further evaluations was described. We will run several experiments with various setups to evaluate system using different speech features described above and findings will be discussed. Since one of the databases defines the task where a searched query can contain several words, we should focus on this problem during the improvement of the system. The baseline system can detect the exact match only. The other improvements from various authors should increase the accuracy of the system performing several normalizations, a fusion of an input data to derive more information, an elimination of non-speech frames to discard needless information, etc.

Chapter 3

Dynamic Time Warping

Dynamic Time Warping (DTW) is a technique for comparing and finding an optimal alignment between two time-dependent sequences of vectors. These sequences are warped in time or in speed to match each other. The goal is to find best mapping between these sequences by warping one or both of them using dynamic programming approach. DTW has been originally used for comparison of speech patterns in automatic speech recognition systems and applied to confront with time-dependent data with time deformation or different speed [8][9].

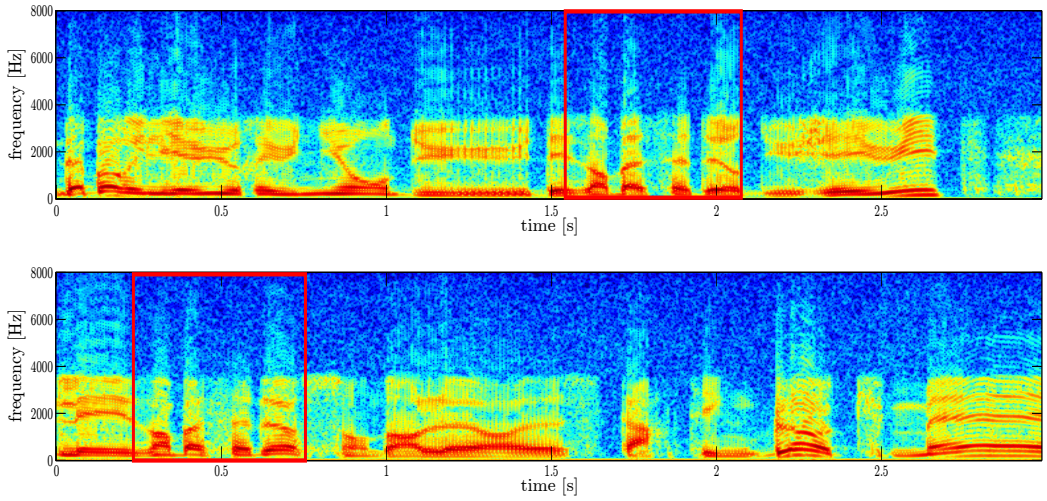


Figure 3.1: *Example of spectrograms for two spoken utterances:* "D´abord il y a eu une reunion des ambassadeurs du G8." *and* "Oui, les ambassadeurs sont dans le starting block.". *The red rectangles mark occurrences of the word* "ambassadeurs" [9].

Besides spoken term detection, DTW has been successfully used in other areas such as data mining, DNA analysis, financial analysis, music and motion analysis and classification or hand-written text recognition [8][9]. In Figure 3.1, two spectrograms of spoken utterances are depicted. The similar areas are marked by red rectangles. The task is to detect these similar areas automatically with a usage of DTW. Figure 3.2 shows alignment of two different one-dimensional signals. Each point is aligned to the closest coincident point from the other sequence.

As mentioned, the objective of DTW is to compare two sequences, find an optimal alignment (Figure 3.2) and return useful information (score value, location of the match,

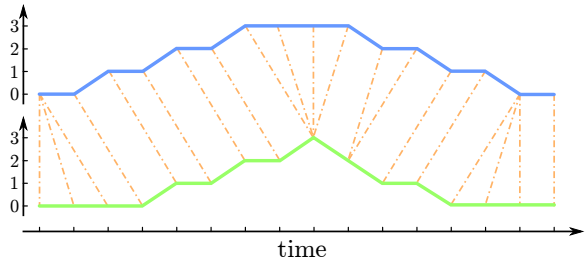


Figure 3.2: *Warping between two different time series. The blue and the green horizontal lines represent two different time series. Each point from one series is optimally aligned with one or more points from the other one and vice versa which allow us to compare even time series with different duration. The warping is shown by the orange dash-dotted vertical lines. Evidently, the warping of series to each other is a non-linear operation* [13][18].

warping path shape, etc.) of this alignment. To describe presented warping more formally, let us consider an utterance $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ as a time-dependent sequence of N vectors and a query $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_M\}$ as a time-dependent sequence of M vectors. All vectors $\mathbf{u} \in \mathbf{U}$ and $\mathbf{q} \in \mathbf{Q}$ have the same dimensionality K. To compare two different sequences of vectors, we need a metric to measure distance between single vectors of these sequences. Let us define the distance metric to compare two vectors $\mathbf{u}$ and $\mathbf{q}$ in general as:

$$d : \mathbf{u} \times \mathbf{q} \to \mathbb{R} \tag{3.1}$$

The distance metrics used in the baseline system were a log-likelihood based on the cosine distance and a log-likelihood based on the dot product.
The log-likelihood based on the cosine distance d_{logcos} is defined by [23]:

$$d_{logcos}(\mathbf{u}, \mathbf{q}) = -\log\left(\frac{\mathbf{u} \cdot \mathbf{q}}{|\mathbf{u}| \cdot |\mathbf{q}|}\right), \tag{3.2}$$

where the expression in parentheses is the cosine similarity. The range of the d_{logcos} is given by the interval $[0, +\infty)$ where 0 denotes identical vectors.

The log-likelihood based on the dot product d_{logdot}, is defined as:

$$d_{logdot}(\mathbf{u}, \mathbf{q}) = -\log(\mathbf{u} \cdot \mathbf{q}), \tag{3.3}$$

where $\cdot$ represents the dot product. The range of the d_{logcos} lies in the interval $[0, +\infty)$ where 0 denotes identical vectors. For both distances, the value of the distance between vectors is lower if vectors are similar to each other and higher if vectors are variant.

3.1 Distance matrix

By calculating distances between all possible query-utterance vectors $\mathbf{u} \in \mathbf{U}$ and $\mathbf{q} \in \mathbf{Q}$, we obtain *distance matrix* $\mathbf{D} \in \mathbb{R}^{N \times M}$ where each cell $D(n, m)$ of the matrix is defined by $d(\mathbf{u}_n, \mathbf{q}_m)$ [8]. Figure 3.3 (on the left) depicts the distance matrix for two real-valued one-dimensional time series (sequences of real numbers) shown in Figure 3.2.

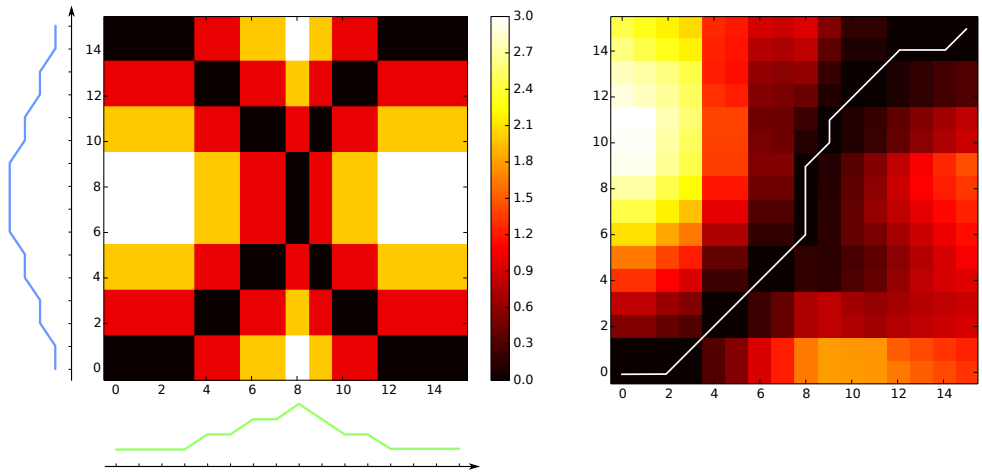


Figure 3.3: *Distance matrix (on the left) for two real-valued sequences from Figure 3.2. The Euclidean distance (4.3) was used to measure distances. The darker colors denote areas where given vectors are similar to each other and the lighter colors symbolize regions of a difference. A cumulative matrix (on the right) corresponds to the distance matrix. The white line represents the optimal warping path* [13][18].

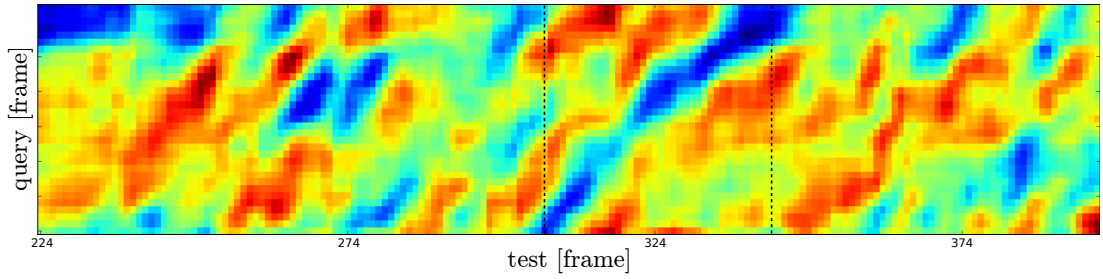


Figure 3.4: *Distance matrix for utterance* "−in his presentation−" *and for query* "present". *The cosine distance (4.1) was used to measure vector distances. The dark blue area between black dotted lines is the match of the query.*

3.2 Cumulative matrix

Cumulative matrix $\mathbf{C}$ accumulates distance values from a distance matrix. Each cell value depends on its predecessor cells (horizontal, vertical and diagonal). The predecessor cell with the lowest value is taken and accumulated with the current cell. The weight factors $(w_d, w_h, w_v) \in \mathbb{R}^3$ serve to favour one of the directions. The standard setup is $(1, \sqrt{2}, 1)$ to treat all directions equally. We used the classical setup $(1, 1, 1)$ to prefer diagonal steps. Formally, cell predecessor *pred* and cumulative matrix $\mathbf{C}$ [8]:

$$pred(n,m) = \arg\min_{n,m} \begin{cases} \mathbf{C}(n-1,m-1) + w_d \cdot \mathbf{D}(n,m) \\ \mathbf{C}(n-1,m) + w_h \cdot \mathbf{D}(n,m) \\ \mathbf{C}(n,m-1) + w_v \cdot \mathbf{D}(n,m) \end{cases} \tag{3.4}$$

$$\mathbf{C}(n,m) = \begin{cases} \mathbf{D}(n,m) & \text{, if } m = 0 \\ \mathbf{C}(n,m-1) + \mathbf{D}(n,m) & \text{, if } n = 0 \\ \mathbf{C}(pred(n,m)) + \mathbf{D}(n,m) & \text{, otherwise} \end{cases} \tag{3.5}$$

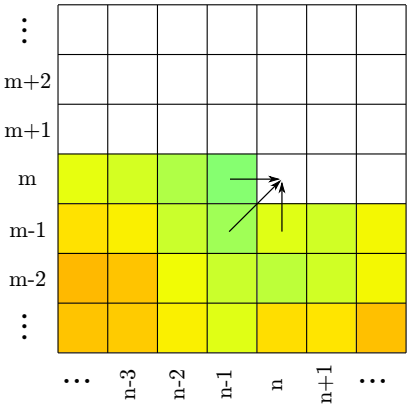


Figure 3.5: *Generation of a cumulative matrix. Possible predecessor cells for the current cell in coordinates* (n,m) *lie in horizontal* $(n-1,m)$*, vertical* $(n,m-1)$ *and diagonal* $(n-1,m-1)$ *direction.*

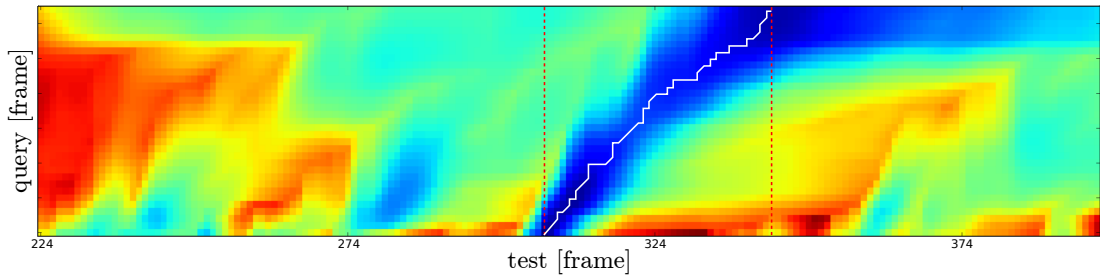


Figure 3.6: *Cumulative matrix for the distance matrix in Figure 3.4. The dark blue area between red dotted lines marks the match of the query. The white line represents the optimal warping path.*

Since the query can appear anywhere in the utterance, the accumulation starts from the origin point $(0,0)$ of distance matrix $\mathbf{D}$ and can reset in the first row $(n,0)$ which lies in time-dependent axis of the utterance. This is performed by simple copying of the first row from distance matrix $\mathbf{D}$ to cumulative matrix $\mathbf{C}$.

In Figure 3.3, a simple cumulative matrix is depicted (on the right). Figure 3.6 shows the cumulative matrix for the previous distance matrix. Last, the cumulative matrix is usually normalized by length.

3.3 Length matrix

Length matrix **L** stores a path length for each cell and is used for the length normalization. During each step of a cumulative matrix calculation, the path length is extended by 1. All cells in the first row $(n, 0)$ are set to 1 due to the fact that a query can start anywhere in the utterance (see Figure 3.7). Formally:

$$\mathbf{L}(n,m) = \begin{cases} 1 & \text{, if } m = 0 \\ \mathbf{L}(pred(n,m)) + 1 & \text{, otherwise} \end{cases} \tag{3.6}$$

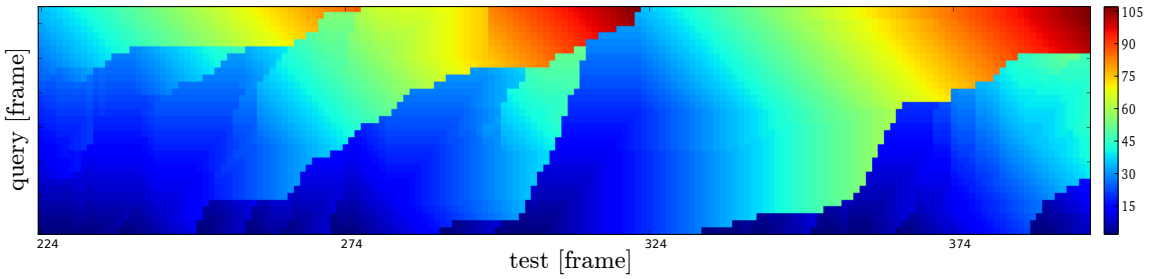


Figure 3.7: *Length matrix for the cumulative matrix in Figure 3.6. Each cell stores the length of a path ending in the cell.*

3.4 Starting-point matrix

Starting-point matrix **S** saves starting points of paths in each cell to avoid further exhaustive computation of paths using a back-tracking. The original starting-point is kept during the computation of a cumulative matrix. Except the first row $(n, 0)$ where the frame number (a matrix column number) is stored. In case a path starts at some point in the first row of the matrix, the frame number (the starting point) is kept and propagated further.
A calculation of starting-point matrix **S** is defined by:

$$\mathbf{S}(n,m) = \begin{cases} n & \text{, if } m = 0 \\ \mathbf{S}(pred(n,m)) & \text{, otherwise} \end{cases} \tag{3.7}$$

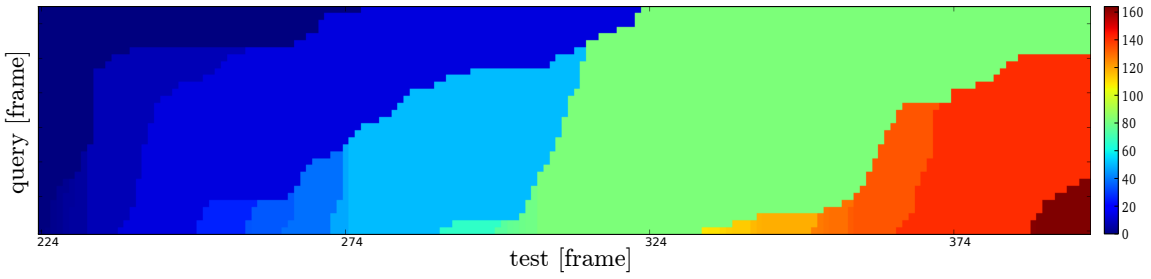


Figure 3.8: *Starting-point matrix for the cumulative matrix in Figure 3.6. Each cell stores the starting point of a path ending in the cell.*

3.5 Optimal path search

Warping path $\mathbf{p}$ is defined as a sequence of points (n, m) following a set of constraints and has a characteristic shape, a length L and a cumulated distortion (score) *dist*. Formally [8]:

$$\mathbf{p} = \{p_1, \ldots, p_L\} = \{(n_1, m_1), \ldots, (n_L, m_L)\}, \tag{3.8}$$

where $(n_l, m_l) \in [0 : N] \times [0 : M]$ *for* $l \in [1 : L]$. A warping path has to satisfy three following conditions [8][9]:

(i) Boundary condition:

$$p_1 = (n_1, 1) \text{ and } p_L = (n_2, M), \text{where } n_1, n_2 \in [0 : N] \tag{3.9}$$

Each path starts in the first row and ends in the last row of matrix $\mathbf{C}$.

(ii) Monotonicity condition:

$$\{(n_1, m_1), (n_2, m_2), \ldots, (n_L, m_L)\} \rightarrow n_1 \leq n_2 \leq \ldots n_L \text{ and } m_1 \leq m_2 \leq \ldots m_L \tag{3.10}$$

Each path is a monotonic function.

(iii) Continuity (step size) condition:

$$p_k - p_{k+1} \in \{(0, 1), (1, 0), (1, 1)\} \text{ for } l \in [1 : L - 1] \tag{3.11}$$

The step is set to adjacent cells only. A skipping of rows or columns is not allowed.

The boundary condition warrants that each path crosses the whole cumulative matrix so the query is enforced to match full-length. The monotonicity condition does not allow the path to get back in time. It is a reflection of the requirement of faithful timing. At last, the step size condition says no vector in sequences $\mathbf{U}$ and $\mathbf{V}$ can be skipped or omitted and there is no possible replication in the alignment. All three conditions are complied during a calculation of a cumulative matrix.

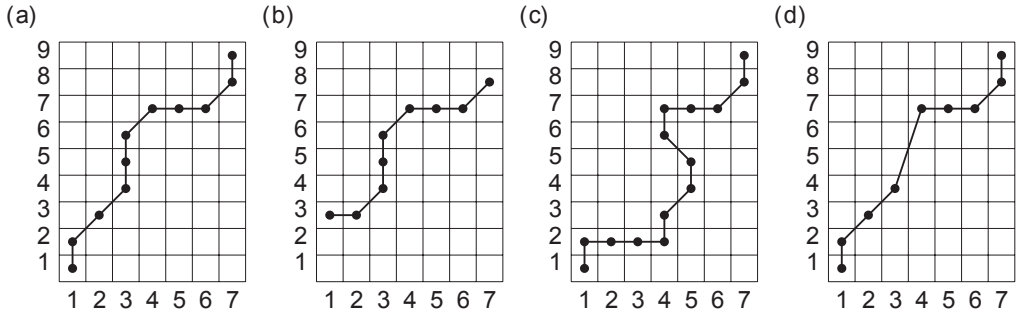


Figure 3.9: *Illustration of paths of index pairs for some sequence U of length* $N = 7$ *and some sequence V of length* $N = 9$. (a) *Admissible warping path satisfying the conditions* (i),(ii), *and* (iii). (b) *Boundary condition* (i) *is violated.* (c) *Monotonicity condition* (ii) *is violated.* (d) *Step size condition* (iii) *is violated* [8].

The total $score_{\mathbf{p}}(\mathbf{U}, \mathbf{V})$ of warping path $\mathbf{p}$ between sequences $\mathbf{U}$ and $\mathbf{V}$ with respect to distance metric d is defined as [8]:

$$score_{\mathbf{p}}(\mathbf{U}, \mathbf{V}) = \sum_{l=1}^{L} d(\mathbf{u}_{n_l}, \mathbf{v}_{m_l}) \tag{3.12}$$

The length normalization of path $\mathbf{p}$ with the total cost $score_{\mathbf{p}}$ and length $L_{\mathbf{p}}$ is defined as:

$$score_{norm_{\mathbf{p}}}(\mathbf{U}, \mathbf{V}) = score_{\mathbf{p}}(\mathbf{U}, \mathbf{V})/L_{\mathbf{p}} \tag{3.13}$$

The optimal warping path $\mathbf{p}_{opt}$ between sequences $\mathbf{U}$ and $\mathbf{V}$ is a warping path with minimal total cost within the cumulate matrix $\mathbf{C}$. Formally [9]:

$$\mathbf{p}_{opt} = \arg\min_{\mathbf{p}} \{score_{norm_{\mathbf{p}}}(\mathbf{U}, \mathbf{V})\} \tag{3.14}$$

Note that all distance metrics used in this work have the same meaning: the lower the value, the closer the vectors (the score is better). That is the reason for searching paths with the lowest score.

To lower the complexity of finding an optimal path $\mathbf{p}_{opt}$, we avoid testing every possible warping path $\mathbf{p}$ between sequences $\mathbf{U}$ and $\mathbf{V}$ in every possible ending point. We used methods based on dynamic programming to construct matrices presented above which allow us to make a searching of paths much simpler. To get several top paths, a distortion profile can be used.

The $profile_{dist}$ stands for the *distortion profile* of cumulative matrix $\mathbf{C}$ represents the last row of $\mathbf{C}$ and is defined by:

$$profile_{dist_{\mathbf{C}}}(n) = \mathbf{C}(n, M) \tag{3.15}$$

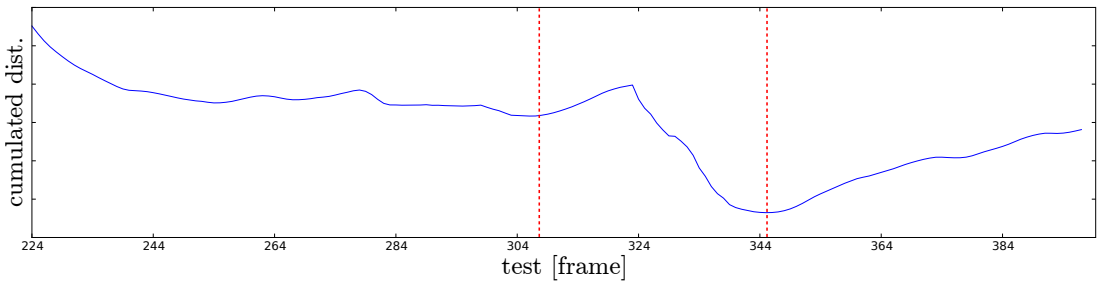


Figure 3.10: *Distortion profile displays values of the last row (n, M) from the cumulative matrix in Figure 3.6. The intersection of the blue curve and the right red dotted line marks the local minimum of distortion profile therefore the path with the best score ends at this frame.*

A distortion profile (see Figure 3.10) stores a cumulated distance of the best possible path for every frame of an utterance. This cumulated distance equals to the normalized score of the optimal warping path ending in given frame (cell of matrix $\mathbf{C}$). As mentioned, the lower value of cumulated distance in distortion profile, the better score of the path. When searching for paths using a distortion profile, we avoid searching for a path in each frame since the location of the best paths is obvious. The end-point of a path is selected from

the distortion profile where the local minimum occurs and the starting point corresponds to the value of the same cell in the starting-point matrix. Note that we lost the information of the path shape, we keep where it starts and ends only. The shape of a path is necessary to set global constraints and to control the route of a path. We did not implemented a path shape control in our algorithm. The simple solution was to filter out warping paths with a slope above half of or below double the query duration. It is a simplification of Itakura parallelogram [6]. The last step is a negation of detection scores so the file containing all detections for a given data sets has the opposite scoring manner: the higher score of the path, the higher confidence of the detection.

3.6 Online length normalization

To normalize accumulated distances according to their length, there are two approaches. The *offline normalization* computes all previously mentioned matrices. During the cumulated matrix computation, it takes into account raw values of the distance matrix in the step of predecessor selection. Last, the cumulate matrix is divided by the length matrix. An optimal path search follows. The other approach, the *online normalization*, performs the division by current path length on-the-fly for every matrix cell calculation to decide which preceding cell is the best to choose. The division is not saved during the calculation, it is performed only to decide the next step. The length normalization is done after all matrices are fully computed as for the offline approach. This leads to prefer longer paths over shorter ones.

The normalized matrices slightly vary for each approach and path shapes are variant as well. We used the online method for normalizing paths by length in our system. In Figure 3.11, the difference between the offline and the online normalized path is depicted.

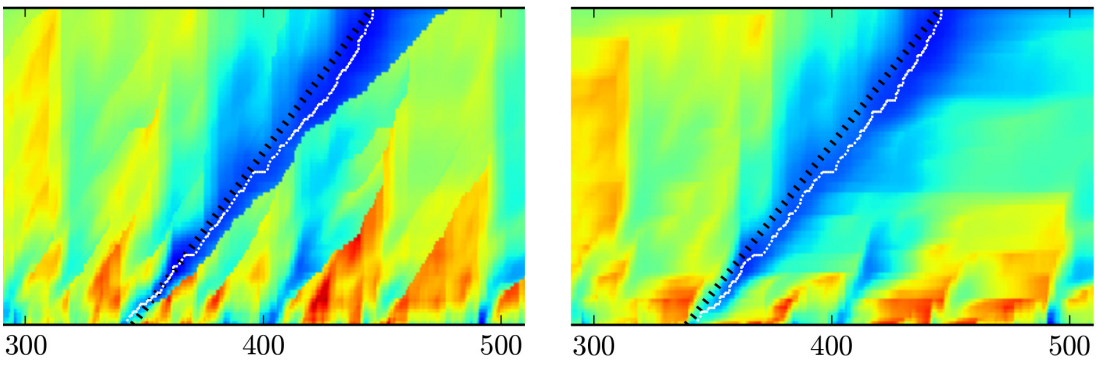


Figure 3.11: *Comparison of the offline (on the left) and the online (on the right) normalization. The online way returns smoother cumulative matrix. The white line represents the back-tracked path. Minor differences between path shapes are visible.*

3.7 Mode normalization of score

The mode normalization is performed to normalize score for each query. The different queries have variant score distributions depending on the ability of the query to be searched. Longer queries are easier to be searched while the shorter ones cause a lot of false alarms. The normalization for each query allows us to use a single threshold maximizing given scoring metrics. The shape of the distribution has a longer tail with bad matching scores and shorter head with good scores. The standard zero mean and unit normalization (4.9) does not take into account this information. The mode is the most appearing value in the set (the peak of a histogram). We subtract the score value in the mode of a histogram from all query scores. The mods of all query histograms are aligned to 0 then. The division by standard deviation for scores larger then the mode follows.

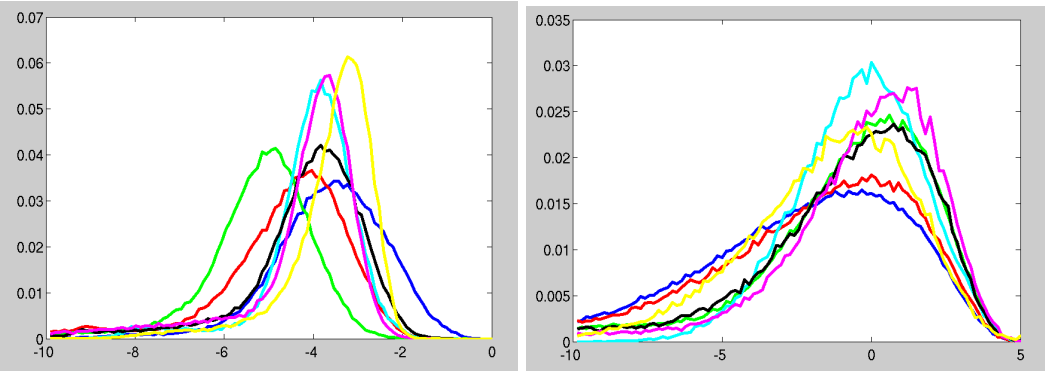


Figure 3.12: *Score distribution histograms for several queries (on the left). The distribution histograms for the same queries after performing the m-normalization (on the right)* [20].

3.8 Baseline experiments

The evaluation of the baseline system was done with the data sets described above. The log-likelihood based on the dot product d_{logdot} and the log-likelihood based on the cosine distance d_{logcos} were used for measuring distances.

Features	d_{logcos}	d_{logdot}
SD CZ POST	0.1319/0.1915	0.1009/0.1788
SD HU POST	0.1196/0.1821	0.0800/0.1557
SD RU POST	0.1571/0.2250	0.1129/0.1985
GP RU POST	0.0268/0.0864	0.0118/0.0603
GP RU BN	0.0799/0.1578	-

Table 3.1: *Results for SWS 2013 development data set. Several features were evaluated with the baseline system. The scoring metric was TWV/UBTWV. The complete tables can be found in appendices.*

3.9 Conclusion

The pattern matching method based on dynamic programming was introduced and described formally. The construction of necessary matrices was presented. The standard DTW includes a computation of a distance and a cumulative matrices followed by back-tracking of an optimal path. In our implementation of the baseline system, we modified the standard approach. A length and a starting-point matrices were added to relive a computationally complex back-tracking at the cost of higher memory consumption. A detection of an optimal path is much simpler using this approach. On-the-fly normalization that prefers longer paths over shorter ones was described. A few experiments were run to get the reference for a comparison with the improved system later. Note that the result for bottleneck features using d_{logdot} is missing. This metric always returns 0.000 and is inadequate for bottleneck features. The baseline system was built by Lukáš Burget.

Chapter 4

My experiments

Several improvements of the baseline system based on literary sources listed in Chapter 2 were made. The most experiments were run on SWS 2013 data set using TWV/UBTWV as the scoring metric. In other cases, the data set and scoring metric is notified.

4.1 Voice Activity Detection

To deal with frames (feature vectors) containing a non-speech signal like silence, a breathing or a noise, *Voice Activity Detection* (VAD) is applied. VAD is performed by discarding vectors where the non-speech posterior is the highest. The remaining feature vectors holding speech are merged together. If the number of remaining vectors is too small, the whole signal is discarded. The threshold was set to 10 speech frames. Shorter queries are harder to detect and cause a lot of false alarms devaluing an overall system performance. We applied VAD on queries only to eliminate silences at the beginning and at the end of speech incurred during manual recording of queries [12].

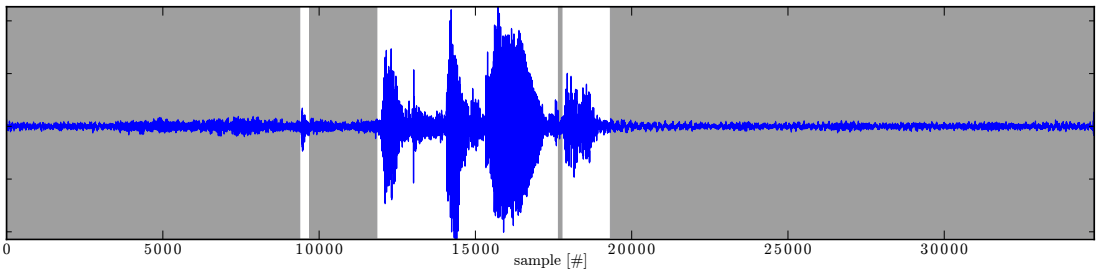


Figure 4.1: *Example of a query audio signal. The grey area marks samples corresponding to feature vectors that were recognized as non-speech and these are discarded after applying of VAD.*

To detect non-speech frames, we extracted 3 sets of phoneme posteriors with *phnrec* [16] phoneme recognizer using Czech, Hungarian and Russian systems. We experimented with a combination of these 3 VADs. The median and the average were performed. In Figure 4.1, an example of VAD applying to speech is shown.

Features	d_{logcos}	d_{logcos} + VAD
SD CZ POST	0.1319/0.1915	0.2246/0.2744
SD HU POST	0.1196/0.1821	0.2125/0.2995
SD RU POST	0.1371/0.2050	0.2315/0.2765
GP RU POST	0.0268/0.0864	0.1030/0.1906
GP RU BN	0.0799/0.1578	0.1493/0.2519

Table 4.1: *Results shows that the application of VAD rapidly increases the performance for all features. VAD almost doubles the score in most cases.*

Experiments shows that non-speech frames cause a lot of false detections. A query containing segments with silence (or noise) matches to silent (noise) segments in an utterance generating needless detections. The application of VAD reduces the number of false detections and boosts the score (see Table 4.1).

Features	d_{corr}	d_{corr} + CZ VAD	d_{corr} + HU VAD	d_{corr} + RU VAD
SD HU POST	0.2202/0.2941	0.4577/0.5417	0.4635/0.5426	0.4567/0.5400
SD RU POST	0.1742/0.2494	0.4348/0.5149	0.4285/0.5114	0.4361/0.5233

Features	d_{corr} + avg VAD	d_{corr} + med VAD
SD CZ POST	0.4398/0.5212	0.4398/0.5212
SD HU POST	0.4577/0.5417	0.4577/0.5417
SD RU POST	0.4341/0.5185	0.4341/0.5185

Table 4.2: *Comparison of VADs generated for different languages. The score enhancement slightly differs but is still significant for all Czech, Hungarian and Russian VADs. The average and the median of these 3 VADs are the same and do not improve the score much.*

After the investigation of different setups for VAD, we found out that VADs based on various languages give us almost the same results. The average and the median of these 3 VADs do not affect the score. From now on, we select Hungarian VAD as the default one and it is applied to all of the following experiments. Complete results for all features can be found in Table A.1 for SWS 2012 and in Table A.2 for SWS 2013.

4.2 Distance metrics

Different metrics[8] for measuring distances between query-utterance vectors were used. The goal was to investigate which distance is the most efficient for different input feature vectors.

The cosine distance d_{cos} is defined as:

$$d_{cos}(\mathbf{u}, \mathbf{q}) = 1 - \frac{\mathbf{u} \cdot \mathbf{q}}{|\mathbf{u}| \cdot |\mathbf{q}|}, \tag{4.1}$$

where $\cdot$ represents the dot product and $|\mathbf{u}|$ stands for the magnitude of vector $\mathbf{u}$. The range of the d_{cos} is given by the interval $[0, 2]$ where 0 denotes identical vectors.

The Pearson product-moment correlation distance d_{corr} is defined by:

$$d_{corr}(\mathbf{u}, \mathbf{q}) = 1 - \frac{(\mathbf{u} - \bar{\mathbf{u}}) \cdot (\mathbf{q} - \bar{\mathbf{q}})}{|(\mathbf{u} - \bar{\mathbf{u}})| \cdot |(\mathbf{q} - \bar{\mathbf{q}})|}, \tag{4.2}$$

where $\bar{\mathbf{u}}$ represents the mean value of vector $\mathbf{u}$. The range of the d_{corr} distance falls into the interval $[0, 2]$ where 0 means identical vectors. Evidently, the only difference between the d_{corr} and the d_{cos} is that the input vectors are mean normalized within the d_{corr}.

The Euclidean distance d_{euc} is defined as:

$$d_{euc}(\mathbf{u}, \mathbf{q}) = \sqrt{\sum_{i=1}^{L} (\mathbf{u}(i) - \mathbf{q}(i))^2}, \tag{4.3}$$

where $\mathbf{u}(i)$ is the i-th element of vector $\mathbf{u}$. The range of the d_{euc} lies in the interval $[0, +\infty)$ where 0 stands for identical vectors.

In addition to these distance metrics, several others were used in experiments without significant results (Bray-Curtis, Canberra, Chebyshev, Mahalanobis, Minkowski and the squared Euclidean).

Features	d_{corr}	d_{cos}	d_{euc}	d_{logcos}	d_{logcos}
SD CZ POST	0.4398/0.5212	0.3739/0.4583	0.1748/0.2506	0.2975/0.3704	0.2923/0.3862
SD HU POST	0.4577/0.5417	0.4079/0.4922	0.2056/0.2899	0.3081/0.3905	0.2916/0.3913
GP RU POST	0.3662/0.4450	0.3336/0.4203	0.1270/0.2063	0.1030/0.1906	0.1087/0.1996
GP RU BN	0.4193/0.5044	0.4208/0.5050	0.1062/0.1732	0.1522/0.2372	-

Table 4.3: *Comparison of different distance metrics. The scoring metric was TWV/UBTWV and SWS 2013 dev database.*

The Pearson correlation was considered as the most robust distance metric regardless the input features as it gave us good results for all types of features (see Table 4.3). Later experiments show that the cosine distance worked better for bottleneck in general and the log-likelihood based on the cosine distance gave us the best results for posteriors (in QUESST 2014 database, see Table A.3).

[8] http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.spatial.distance.cdist.html

4.3 Principle Component Analysis

Principle Component Analysis (PCA) is a method converting the data with a correlation into a set of linearly uncorrelated data. A covariance matrix stores the covariance between vector elements of the input data. Eigenvectors of a covariance matrix defines the coordinate bases where the data are decorrelated. Eigenvalues of a covariance matrix represents the variability in each dimension. The decorrelated data have a diagonal covariance matrix. A projection of several bases with a high variability can be performed to decrease dimensionality of the data with a low information loss. The transformed data can be optimally reconstructed with a low mean square error [3][16].

We analysed all input features and performed PCA on each of the sets to decorrelate feature vectors elements. The results can be found on attached DVD. An example of an investigation of adapted bottleneck features is in Figure 4.2.

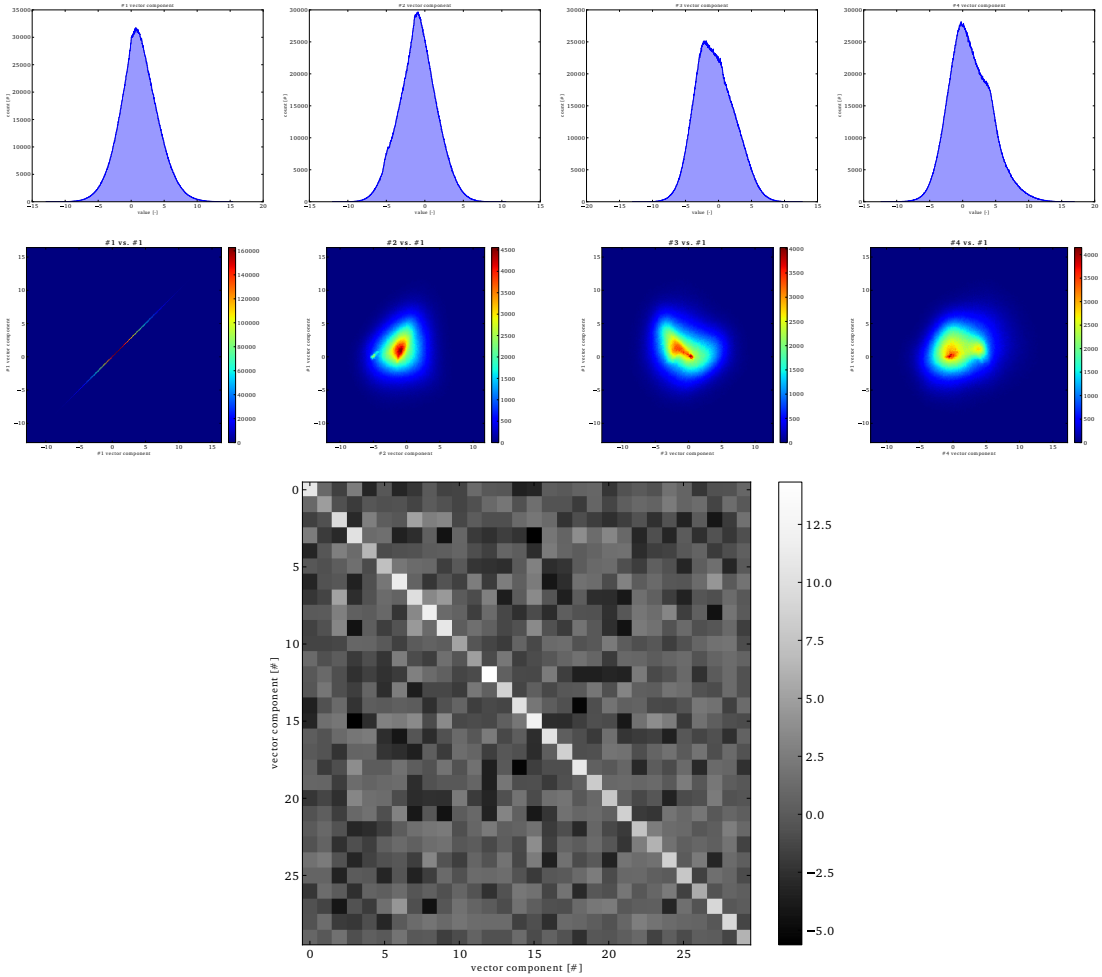


Figure 4.2: *Histograms for the first 4 out of 30 elements of bottleneck feature vectors with adaptation (on the top). The correlation between the first vector element and the following ones (in the middle). The covariance matrix for adapted bottleneck features (in the bottom).*

Features	d_{corr}	d_{corr} + PCA
GP RU BN	0.3998/0.4829	0.3696/0.4519

Features	d_{corr}	d_{corr} + PCA
SD HU POST	55.08/0.518	46.48/0.611
GP RU BN	51.90/0.539	43.03/0.618

Table 4.4: *PCA transformation of input features does not improve the overall score. Top table: SWS 2013 dev set and TWV/UBTWV. Bottom table: QUESST 2014 dev set and MTWV/C^{min}_{nxe} for T1 query type.*

Since PCA transformation does not bring any interesting results, we do not apply it in further experiments.

4.4 0/1 matrix normalization

The *0/1 normalization* is performed on the distance matrix. This matrix is normalized with regard to utterance **U**. Cell values of the matrix are comprised between 0 and 1. Distance matrices have the same range regardless the acoustic condition or the speaker in the utterance. Therefore, an optimal path should have a score close to zero. Formally [12]:

$$d_{norm}(\mathbf{u}, \mathbf{q}) = \frac{d_x(\mathbf{u}, \mathbf{q}) - d_{min}(\mathbf{q})}{d_{max}(\mathbf{q}) - d_{min}(\mathbf{q})}, \tag{4.4}$$

where $d_x(\mathbf{u}, \mathbf{q})$ is one of the presented distance metrics and:

$$d_{min}(\mathbf{q}) = \min_{\mathbf{u} \in \mathbf{U}} d_x(\mathbf{u}, \mathbf{q}) \tag{4.5}$$

$$d_{max}(\mathbf{q}) = \max_{\mathbf{u} \in \mathbf{U}} d_x(\mathbf{u}, \mathbf{q}) \tag{4.6}$$

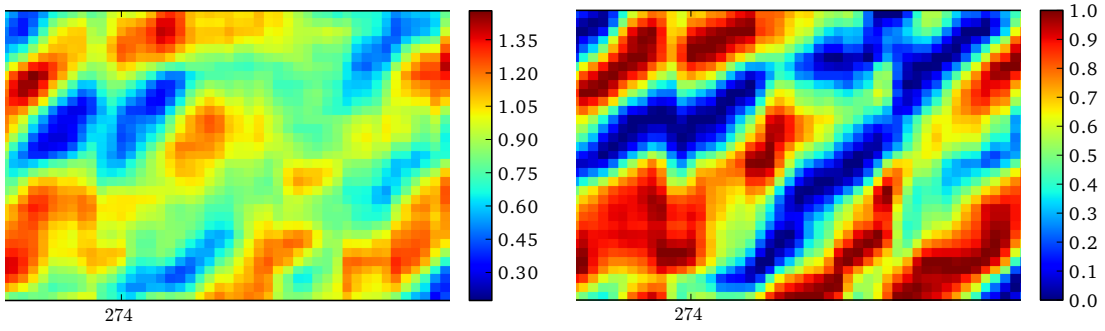


Figure 4.3: *Distance matrix calculated using the correlation distance 4.2 (on the left). The same distance matrix after 0/1 normalization (on the right). The color contrast shows that all cell values are comprised between value 0 and 1.*

Features	d_{cos}	d_{cos} + 0/1 n.	d_{euc}	d_{euc} + 0/1 n.
GP RU BN	0.4208/0.5050	0.3272/0.4116	0.1062/0.1732	0.0678/0.1427

Features	d_{corr}	d_{corr} + 0/1 n.
SD HU POST	0.4577/0.5417	0.3770/0.4598

Table 4.5: *0/1 transformation of input features does not improve the overall score. SWS 2013 dev set and TWV/UBTWV were used.*

No improvements were achieved by 0/1 normalizing the distance matrix. We do not normalize this way in later experiments.

4.5 Fusion using concatenation of features

The concatenation of extracted phoneme posteriors or bottlenecks was used as features. The vectors were simply stacked on each other to create a large feature vector. In particular, we created a concatenation of features from Czech, Hungarian and Czech phoneme decoders and all 7 languages for GP decoders. We tried several combinations and ended up fusing Czech, Portuguese, Russian and Spanish bottlenecks for QUESST database.

Features	d_{corr}
SD HU POST (best single)	0.4577/0.5417
SD fusion POST (CZ+HU+RU)	0.4599/0.5439
GP RU POST	0.3662/0.4450
GP fusion POST (7langs)	0.4049/0.4896
GP RU BN	0.4193/0.5044
GP fusion BN (7langs)	0.5122/0.5946

Table 4.6: *Fusion of input features improves score in all cases. The scoring metric was TWV/UBTWV and SWS 2013 dev database.*

The concatenation of input features works well. A fusion of the best single system with some other worse ones improves the score only a little in general. However, fusing several average systems enhances the score significantly.

4.6 Fusion using parallel tokenizers

The parallel tokenizer system includes several different tokenizers which are expected to complement each other. Each tokenizer extract features for input query and utterance and computes the distance matrix. Output matrices from all tokenizers are merged into one distance matrix then and DTW is performed [26]. The system is depicted in Figure 4.4. In our implementation, we used already extracted features from decoders to compute distance matrices and then these were merged together.

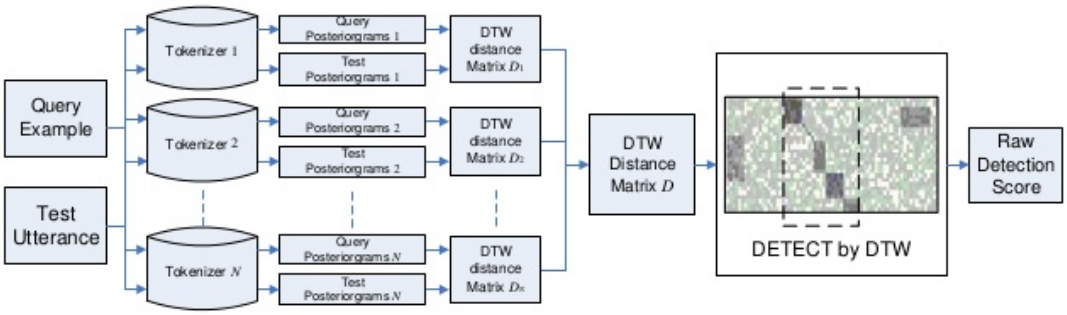


Figure 4.4: *Parallel tokenizer system* [26].

Features	d_{corr}
GP fusion BN (concat)	0.5122/0.5946
GP fusion BN (parallel)	0.5120/0.5924

Table 4.7: *Fusion using parallel tokenizers yields results similar to the previous approach with the concatenation.*

Experiments showed that the concatenation of features and parallel systems returns similar score. The concatenation consumes more memory as all feature vectors are read simultaneously. The parallel tokenizers are more computationally complex as matrices are computed in parallel. We decided to use the concatenation for further evaluations as the default one.

4.7 Summing of phoneme-states

The decoders returns phoneme-state probabilities of 3 states for each of phoneme units at each frame. Adding the probabilities for each unit can be defined formally as [12]:

$$p_i(t) = \sum_{\forall s} p_{i,s}(t), \tag{4.7}$$

where $p_{i,s}(t)$ is the probability of state s of unit i at frame t.

Features	d_{corr}	d_{corr} + SUM
SD CZ POST	0.4398/0.5212	0.3757/0.4568
SD HU POST	0.4577/0.5417	0.4248/0.5068

Table 4.8: *Summing of states used for phoneme-state posteriors.*

Reducing the size of input feature vectors by summing the states probably leads to information loss and thus the score decreases. The summing was not used during later experiments.

4.8 Scaling of score

Each query match score is normalized by the following formula [25]:

$$score_{norm} = e^{\left(\frac{-score}{\beta}\right)}, \tag{4.8}$$

where *score* corresponds to cumulated distance from the distance profile and β is the scaling factor. To calibrate the score distribution for each query, the standard zero mean and unit normalization was used:

$$score_{calib} = \frac{score_{norm} - \mu_{\mathbf{q}}}{\sigma_{\mathbf{q}}}, \tag{4.9}$$

where $\mu_{\mathbf{q}}$ stands for mean and $\sigma_{\mathbf{q}}$ is a variance of all (or top) scores for query $\mathbf{q}$.

β	3	4	5	6	7
GP CZ POST	0.3540	0.3543	0.3537	0.3532	0.3531

Table 4.9: *Results for different scaling factor β and the number of top scores = 400. The score is MTWV.*

# of top scores	50	100	200	400	800	1600	3200	6400
GP CZ POST	0.3132	0.3423	0.3525	0.3537	0.3544	0.3538	0.3532	0.3524

Table 4.10: *Results for different number of top scores and scaling factor $\beta = 5$. The score is MTWV.*

Experiments shows the highest score was achieved for $\beta = 4$ and 800 top scores. However, the scaling does not outperformed the mode normalization so we left the mode as the default score normalization.

4.9 Type 3 of query

To deal with Type 3 query defined in QUESST database, we divided the matrices into sub-bands. The searching for a query match was done in each sub-band separately. The number of sub-bands we experimented with was set to 2, 3 and 4 and this number follows possible number of words in a term. The width of sub-bands is equal and uniform. We expect that 2 words in a term are separated somewhere in the middle (which is not guaranteed by definition). We search for the single word in each sub-band then. The best detection from each sub-band is taken and scores are summed up. If the query contains only one word, the best detections from all sub-bands are linked to each other (see Figure 4.5). If the query consists of 2 words, each word is found elsewhere but the term is detected successfully. In the similar way, the detection works for more than 2 sub-bands.

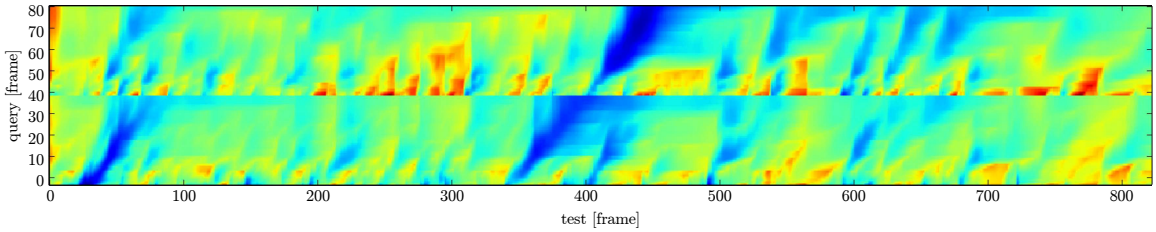


Figure 4.5: *Cumulative matrix is split in half and contains two sub-bands.*

Features	d_{corr} + 1 band	d_{corr} + 2 bands	d_{corr} + 3 bands
SD CZ POST	33.16(51.44/26.88/9.07)	28.89(38.04/19.34/20.58)	23.05(31.41/13.74/18.04)

Table 4.11: *Results for splitting of matrices into sub-bands to enhance search of query Type 3. The scoring metric is MTWV. The numbers stand for overall score and Type 1/Type 2/Type 3 scores in parentheses.*

As can be seen in Table 4.11, the overall score is decreasing with the growing number of sub-bands. On the other hand, the score for query Type 3 doubles for more than 1 sub-bands which was our goal of this improvement. The problem is the degradation of scores for Type 1 and Type 2 queries. Since this upgrade causes worse score in general, this division of matrices was not used during final evaluations.

4.10 Conclusion

We described all investigated improvements or upgrades of the baseline system in detail. The first was the application of VAD. After experimenting with combinations of different VADs, the single VAD based on Hungarian language was chosen. We tried various distances. The cosine distance was the best for bottleneck features. The log-likelihood of the cosine distance worked the best with phoneme posteriors. However, the Pearson correlation provided good results for all used features so it was considered the most robust distance metric regardless the input. Next upgrade, PCA transformation, did not improve the score. Neither worked the normalization of a distance matrix. Both the concatenation and parallel tokenizers returned very similar and impressive results. The sum of phoneme states provided worse score. The scaling did not outperformed the baseline mode normalization. Last, we experimented with Type 3 of query. We improved the score for the given type but the overall score decreased. An investigation of this phenomenon could be a part of future research and experiments.

Chapter 5

Evaluation system

In this chapter, we present the system participating in QUESST task in MediaEval 2014. This system was build by Igor Szöke and Lukáš Burget and subsystems based on DTW were modified by the author of this thesis. The datasets and scoring metrics for evaluation are presented. The description of the system follows. The normalization and fusion of score is outlined. Last, the results are discussed. This chapter is adopted from [22].

5.1 Dataset and scoring metrics

The QUESST 2014 data set is described in detail in Chapter 2. The primary scoring metric used for evaluation was normalized cross entropy C_{nxe}, the secondary metric was MTWV, both presented in Chapter 2.

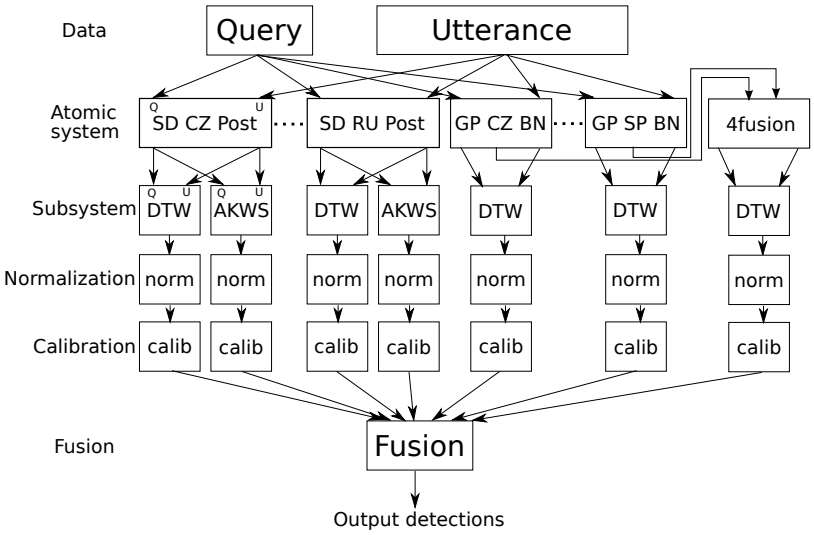


Figure 5.1: *BUT*[9] *Query-by-Example system. Q means queries as an input, U stands for utterances as an input SD means SpeechDat atomic systems where the output are phoneme-state posteriors, GP stands for GlobalPhone atomic systems where the output are bottleneck features* [22].

[9] Brno University of Technology

5.2 System overview

BUT QbE system is depicted in Figure 5.1. The system consists of POST and BN extractors called atomic system. The extraction is outlined in Chapter 2. We used 7 atomic systems: 3× using phoneme-state posteriors, 4× bottleneck features. The phoneme posteriors extractors were trained on SD database using Czech, Hungarian and Russian languages. The bottlenecks extractors were trained on GP database using Czech, Portuguese, Russian and Spanish languages.

Two types of subsystems were exploited: first is based on AKWS (more in [19]) and the other on DTW. The input of each subsystem is feature vectors and the output is a set of detections.

5.3 Score normalization and calibration

The mode normalization was applied to normalize scores per query. The very best detection was selected for each query-utterance pair. The calibration was performed using the binary logistic regression. We used additional information (sideinfo) for the calibration. In addition to mode normalized score, the number of phonemes, the log of number of phonemes, the number of speech frames, the log of the number of speech frames, the average of log-posterior of speech frames obtained from VAD and the LID i-vector score (for more details see [22]).

5.4 Fusion

The fusion (concatenation) of features (denoted as 4fusion) is described in Chapter 4. We concatenated bottlenecks of 4 GP extractors.

The other fusion using subsystems output fuses normalized and calibrated scores with the binary logistic regression linear classifier.

5.5 Results

In Table A.3, a comparison of presented distance metrics on the development dataset is shown. The 7 atomic systems for fusion were selected during experiments and adding extra systems does not improve overall score significantly. The best single system using features from a recognizer trained on Czech language matches Czech and Slovak part of the database which explains its highest accuracy. The overall evaluation results are shown in Table 5.1.

System	C_{nxe} / C_{nxe}^{min}
BUT 4fusion	0.473 / 0.466
BUT GP CZ BN	**0.536 / 0.528**
NTU-NPU-I2R	0.602 / 0.598
EHU	0.621 / 0.599
SPL-IT	0.659 / 0.508
CUHK	0.683 / 0.659
IIIT-H	0.921 / 0.812

Table 5.1: *QUESST 2014 results for the evaluation dataset. The best systems for 6 out of 15 registered participants are listed.* C_{nxe} *and* C_{nxe}^{min} *score for each system is presented. The winning system was BUT 4fusion system. The single system based on DTW and developed by the author (shown in bold) outperformed other participants according to* C_{nxe} *metric.*

Chapter 6

Conclusions and contribution

The aim of this thesis was to investigate keyword spotting methods where the query is provided as an audio example. The requirement was to suggest and implement new techniques to improve the QbE system.

The theoretical part of this thesis focused on outlining the bases of spoken term detection and keyword spotting methods. The textual STD was presented and the reasons for usage of QbE where the query is entered as a speech sample were explained. The QbE STD procedure was split into basic blocks and each block was detailed.

The data sets used as an input for implemented system were presented. The extractors based on artificial neural networks exploited to generate speech features were described. We used the phoneme-state posterior probabilities and the bottleneck features.

The definition of scoring metrics followed. The well-known TWV metric and associated ATWV, UBTWV and MTWV were used. The newly introduced metric called normalized cross entropy was defined. We found out that MTWV and cross entropy are corresponding metrics to each other: both rise or fall in similar rate for different experiments.

The pattern matching method DTW was described. The algorithm consists of necessary matrices used for the standard DTW approach. In addition to these matrices, we used other ones to simplify the searching of detections at the cost of higher memory consumption. We also modified the way of the length normalization. We run experiments to set the reference for further improvement and upgrade of the baseline system. We found out that the logarithm based on the dot product did not work for bottleneck features in any experiment.

The practical part of this thesis consists of several modifications of the baseline system followed by experimental testing. These modifications were based on the related work. One of the best improvements was achieved by applying VAD to discard non-speech fragments from speech features. We chose the VAD based on Hungarian language as it gave us the best results during experiments in long term. The different distance metric were tested. The best score for bottleneck features provided the cosine distance. The best results for phoneme-state posteriors yielded the log-likelihood of the cosine distance. The general metric was considered the Pearson correlation distance as it provided good results regardless the input features. To assess the best features, the posteriors extracted on SD database and Hungarian language returned very good results. The bottlenecks extracted on GP database and Czech languages performed well. The PCA transformation lead to decrease of score as well as the normalization of a distance matrix. The next impressive improvement was achieved by the concatenation of input features. We concluded a superiority of bottleneck features for this fusion. The other method using parallel tokenizers returns similar results.

We chose the first method due to its simple implementation. Several experiments were run to deal with the query Type 3 but with no applicable results. From the baseline to the final improved system, we achieved big increase of the system performance.

The implemented system was a part (subsystem) of more complex system participating in QUESST evaluations in MediaEval 2014. The whole system using the fusion outperformed all the other participants. The single best system designed by the author achieved also excellent results and scored the second.

6.1 Publications

The results of this work have been presented at Excel@FIT[10] 2015 Student conference of innovations, technologies and science in IT held by Faculty of Information Technology, Brno University of Technology [18]. A printed version of the presented poster (A1 size, in Czech language) is attached to this thesis. The overall system description has been published in *BUT QUESST 2014 System Description* paper [21] in MediaEval QUESST[11] 2014 and a bit more detailed version is in *Coping with Channel Mismatch in Query-by-Example - BUT QUESST 2014* paper [22] in ICASSP[12] 2015.

6.2 Future work

The future work could include the investigation of Type 2 and Type 3 of query. The task is to build a general system which could detect all three types in one run or search for each type separately and then combine the results in some clever way. Our approach was not adequate.

The features are the next thing to focus on in the future since the DTW relies on the quality of its input. The generation of better features could lead to increase of the system performance.

[10] http://excel.fit.vutbr.cz/
[11] http://www.multimediaeval.org/mediaeval2014/quesst2014/
[12] http://icassp2015.org/

Bibliography

[1] Anguera, X., Metze, F., Buzo, A., Szöke, I., and Rodriguez Fuentes, L. J. The Spoken Web Search Task. In *Proceedings of the Mediaeval 2013 Evaluation Workshop* (2013).

[2] Anguera, X., Rodriguez-Fuentes, L. J., Szöke, I., Buzo, A., and Metze, F. Query by Example Search on Speech at MediaEval 2014. In *Proceedings of the Mediaeval 2014 Workshop* (2014).

[3] Burget, L. *Complementarity of Speech Recognition Systems and System Combination*. PhD thesis, Brno, Brno University of Technology, Faculty of Information Technology, 2004.

[4] Fapšo, M. *Query-by-Example Spoken Term Detection*. PhD thesis, Brno, Brno University of Technology, Faculty of Information Technology, 2014.

[5] Grézl, F., and Karafiát, M. Hierarchical Neural Net Architectures for Feature Extraction in ASR. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)* (2010), vol. 2010, International Speech Communication Association, pp. 1201–1204.

[6] Itakura, F. Readings in Speech Recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990, ch. Minimum Prediction Residual Principle Applied to Speech Recognition, pp. 154–158.

[7] Metze, F., Anguera, X., Barnard, E., Davel, M., and Gravier, G. The Spoken Web Search Task at MediaEval 2012. In *Proceedings of ICASSP 2013* (Vancouver, Canada, 2013), pp. 8121–8125.

[8] Müller, M. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[9] Muscariello, A. *Variability Tolerant Discovery of Arbitrary Repeating Patterns in Audio Data*. Theses, Université Rennes 1, Jan. 2011.

[10] Pollák, P., Černocký, J., Boudy, J., Choukri, K., Heuvel, H. V. D., Vicsi, K., Virag, A., Siemund, R., Majewski, W., Staroniewicz, P., Tropf, H., Kochanina, J., Ostroukhov, E., Rusko, M., and Trnka, M. SpeechDat(E)- Eastern European Telephone Speech Databases. In *Proc. of XLDB 2000, Workshop on Very Large Telephone Speech Databases* (2000).

[11] Rodriguez Fuentes, L. J., and Penagarikano, M. MediaEval 2013 Spoken Web Search Task: System Performance Measures.

[12] Rodriguez Fuentes, L. J., Varona, A., Penagarikano, M., Bordel, G., and Diez, M. High-performance Query-by-Example Spoken Term Detection on the SWS 2013 Evaluation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (Florence, Italy, 4-9 May 2014).

[13] Salvador, S., and Chan, P. Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intell. Data Anal. 11*, 5 (Oct. 2007), 561–580.

[14] Schultz, T. GlobalPhone: A Multilingual Speech and Text Database Developed at Karlsruhe University. In *Proceedings od the ICSLP* (2002), pp. 345–348.

[15] Schultz, T., Thang Vu, N., and Schlippe, T. GlobalPhone: A Multilingual Speech and Text Database in 20 Languages. In *Proceedings od the ICASSP* (2013), IEEE, pp. 8126–8130.

[16] Schwarz, P. *Phoneme Recognition Based on Long Temporal Context*. PhD thesis, Brno, Brno University of Technology, Faculty of Information Technology, 2009.

[17] Skácel, M. *Searching Acoustic Patterns in Speech Data without Recognition*. Bachelor's thesis, Brno, Brno University of Technology, Faculty of Information Technology.

[18] Skácel, M. Query-by-Example Spoken Term Detection. In *Proceedings of Excel@FIT 2015* (2015), pp. 420–425.

[19] Szöke, I. *Hybrid Word-subword Spoken Term Detection*. PhD thesis, Brno, Brno University of Technology, Faculty of Information Technology, 2010.

[20] Szöke, I., Burget, L., Grézl, F., and Ondel, L. BUT SWS 2013 - Massive Parallel Approach. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop* (2013), vol. 2013, CEUR-WS.org, pp. 1–2.

[21] Szöke, I., Skácel, M., and Burget, L. BUT QUESST 2014 System Description. In *Proceedings of the Mediaeval 2014 Workshop* (2014).

[22] Szöke, I., Skácel, M., Burget, L., and Černocký, J. Coping with Channel Mismatch in Query-by-Example - BUT QUESST 2014. In *Proceedings of ICASSP 2015* (2015), IEEE Signal Processing Society, pp. –.

[23] Tejedor, J., Fapšo, M., Szöke, I., Černocký, J. H., and Grézl, F. Comparison of Methods for Language-dependent and Language-independent Query-by-Example Spoken Term Detection. *ACM Trans. Inf. Syst. 30*, 3 (Sept. 2012), 18:1–18:34.

[24] Veselý, K., Karafiát, M., Grézl, F., Janda, M., and Egorova, E. The Language-independent Bottleneck Features. In *Proceedings of IEEE 2012 Workshop on Spoken Language Technology* (2012), IEEE Signal Processing Society, pp. 336–341.

[25] Wang, H., and Lee, T. The CUHK Spoken Web Search System for MediaEval 2013. In *MediaEval* (2013), M. A. Larson, X. Anguera, T. Reuter, G. J. F. Jones, B. Ionescu, M. Schedl, T. Piatrik, C. Hauff, and M. Soleymani, Eds., vol. 1043 of *CEUR Workshop Proceedings*, CEUR-WS.org.

[26] Wang, H., Lee, T., Leung, C.-C., Ma, B., and Li, H. Using Parallel Tokenizers with DTW Matrix Combination for Low-resource Spoken Term Detection. In *ICASSP* (2013), IEEE, pp. 8545–8549.

Appendix A

Appendices

A.1 DVD contents

The enclosed DVD contains pdf version of this thesis, latex source files for this thesis, all figures used in this thesis, the poster (all in *doc* folder), results for the input features analysis (*histograms* folder), PCA images (*PCA* folder), python and shell scripts (*scripts* folder). The directory structure of the DVD is following:

```
/
+- doc
|  +- src_latex/
|  |  +- fig/
|  +- thesis.pdf
+- histograms
+- pca
+- scripts
```

A.2 Scripts

The attached scripts are used for DTW search, score normalization and calibration, plotting of histograms, matrices, covariance matrices, combination of VAD and many others. Scripts have the instructions for use in the header of the file.

	SWS 2012 dev	
	TWV	UBTWV
GP CZ BN + adapt + logcos + vad	0.1794	0.1187
GP CZ BN + logcos + vad	0.1905	0.1262
GP CZ POST + logdot + vad	0.1644	0.0894
GP EN BN + adapt + logcos + vad	0.0701	0.0468
GP EN BN + logcos + vad	0.0878	0.0353
GP EN POST + logdot + vad	0.0417	0.0299
GP GE BN + adapt + logcos + vad	0.0526	0.0224
GP GE BN + logcos + vad	0.1169	0.0670
GP GE POST + logdot + vad	0.0253	0.0076
GP PO BN + adapt + logcos + vad	0.1859	0.1161
GP PO BN + logcos + vad	0.1829	0.1349
GP PO POST + logdot + vad	0.0794	0.0509
GP RU BN + adapt + logcos + vad	0.1584	0.0889
GP RU BN + logcos + vad	0.1611	0.1255
GP RU POST + logdot + vad	0.0372	0.0171
GP SP BN + adapt + logcos + vad	0.1691	0.1216
GP SP BN + logcos + vad	0.1552	0.1195
GP SP POST + logdot + vad	0.0944	0.0454
GP TU BN + adapt + logcos + vad	0.1304	0.0776
GP TU BN + logcos + vad	0.1089	0.0659
GP TU POST + logdot + vad	0.0536	0.0256
GP VI BN + adapt + logcos + vad	0.1349	0.0930
GP VI BN + logcos + vad	0.0875	0.0713
GP VI POST + logdot + vad	0.0776	0.0465

Table A.1: *Results of the baseline system for SWS 2012 data set.*

	SWS 2013 dev		SWS 2013 eval	
	TWV	UBTWV	TWV	UBTWV
SD CZ POST + logcos + vad	0.2246	0.2744	-	-
SD CZ POST + logdot + vad	0.2038	0.2946	-	-
SD HU POST + logcos + vad	0.2125	0.2995	-	-
SD HU POST + logdot + vad	0.1675	0.2790	-	-
SD RU POST + logcos + vad	0.2315	0.2765	-	-
SD RU POST + logdot + vad	0.2500	0.3379	-	-
GP CZ BN + adapt + logcos + vad	0.1898	0.2858	0.1250	0.2234
GP CZ BN + logcos + vad	0.2705	0.3724	0.2099	0.3166
GP CZ POST + logdot + vad	0.3154	0.4170	0.2474	0.3719
GP EN BN + adapt + logcos + vad	0.0787	0.1905	0.0383	0.1496
GP EN BN + logcos + vad	0.1164	0.2262	0.0885	0.1928
GP EN POST + logdot + vad	0.1041	0.2212	0.0767	0.1831
GP GE BN + adapt + logcos + vad	0.0557	0.1623	0.0339	0.1337
GP GE BN + logcos + vad	0.1179	0.2252	0.0919	0.1839
GP GE POST + logdot + vad	0.0850	0.2081	0.0593	0.1667
GP PO BN + adapt + logcos + vad	0.2100	0.3094	0.1527	0.2507
GP PO BN + logcos + vad	0.1770	0.2681	0.1011	0.1908
GP PO POST + logdot + vad	0.1039	0.2259	0.0609	0.1822
GP RU BN + adapt + logcos + vad	0.1514	0.2507	0.1079	0.2058
GP RU BN + logcos + vad	0.1493	0.2519	0.0884	0.1943
GP RU POST + logdot + vad	0.0369	0.1442	0.0215	0.1241
GP SP BN + adapt + logcos + vad	0.1995	0.2980	0.1286	0.2259
GP SP BN + logcos + vad	0.1508	0.2666	0.0935	0.1941
GP SP POST + logdot + vad	0.1521	0.2697	0.1033	0.2155
GP TU BN + adapt + logcos + vad	0.1364	0.2413	0.1131	0.2147
GP TU BN + logcos + vad	0.0971	0.1924	0.0642	0.1573
GP TU POST + logdot + vad	0.0676	0.1680	0.0353	0.1456
GP VI BN + adapt + logcos + vad	0.1460	0.2404	0.1138	0.2157
GP VI BN + logcos + vad	0.0854	0.1941	0.0423	0.1402
GP VI POST + logdot + vad	0.1326	0.2299	0.0811	0.1894

Table A.2: *Results of the baseline system for SWS 2013 data set.*

Approach	corr	cos	euc	logcos	logdot
SD CZ POST	0.687(0.534/0.761)	0.768(0.633/0.829)	0.852(0.806/0.863)	**0.649**(0.453/0.724)	0.658(0.460/0.735)
	33.35(51.71/28.37)	25.14(40.43/20.07)	11.26(18.23/9.24)	29.51(52.78/23.51)	29.42(52.61/23.72)
SD HU POST	**0.646**(0.505/0.711)	0.712(0.584/0.767)	0.805(0.731/0.827)	0.679(0.510/0.752)	0.691(0.523/0.765)
	37.26(55.36/31.34)	30.37(46.77/26.03)	15.43(25.18/12.16)	28.94(47.85/25.35)	28.96(48.28/24.56)
SD RU POST	0.653(0.509/0.707)	0.706(0.557/0.771)	0.789(0.712/0.820)	**0.652**(0.495/0.720)	0.662(0.510/0.727)
	36.60(54.44/31.77)	31.81(48.47/27.04)	17.87(29.57/14.94)	31.00(51.09/27.73)	30.77(50.13/27.47)
GP CZ BN	0.593(0.435/0.654)	**0.585**(0.425/0.651)	0.777(0.672/0.809)	0.722(0.601/0.771)	-
	42.71(61.32/37.57)	42.30(60.49/38.72)	14.83(28.16/12.14)	24.93(42.07/21.35)	-
GP PO BN	0.659(0.536/0.709)	**0.650**(0.522/0.707)	0.882(0.830/0.900)	0.819(0.750/0.847)	-
	36.17(51.31/30.33)	**36.45**(52.03/31.20)	4.53(11.19/2.67)	13.43(25.32/11.36)	-
GP RU BN	0.668(0.533/0.726)	**0.658**(0.516/0.723)	0.862(0.800/0.882)	0.814(0.726/0.848)	-
	35.11(51.84/30.45)	**35.65**(52.98/32.04)	8.53(16.91/7.88)	13.54(26.76/8.61)	-
GP SP BN	0.673(0.558/0.716)	**0.663**(0.540/0.715)	0.849(0.773/0.867)	0.822(0.741/0.853)	-
	35.56(50.28/31.62)	35.33(50.52/31.85)	10.81(19.03/10.42)	12.57(24.74/10.18)	-
GP CZ+PO+RU+SP BN	0.586(0.432/0.649)	**0.579**(0.418/0.652)	0.761(0.671/0.786)	0.713(0.601/0.758)	-
4fusion	44.63(60.99/39.35)	**44.75**(61.83/40.05)	18.89(32.84/18.44)	25.22(41.04/22.85)	-

Table A.3: *Results of the improved system for QUESST 2014 database. The score in front of parenthesis is score for all queries on dev set, the number in parenthesis for Type 1/Type 2 query. The scoring metric stands for* C_{nxe}^{min} *(top) and* $MTWV$ *(bottom).*

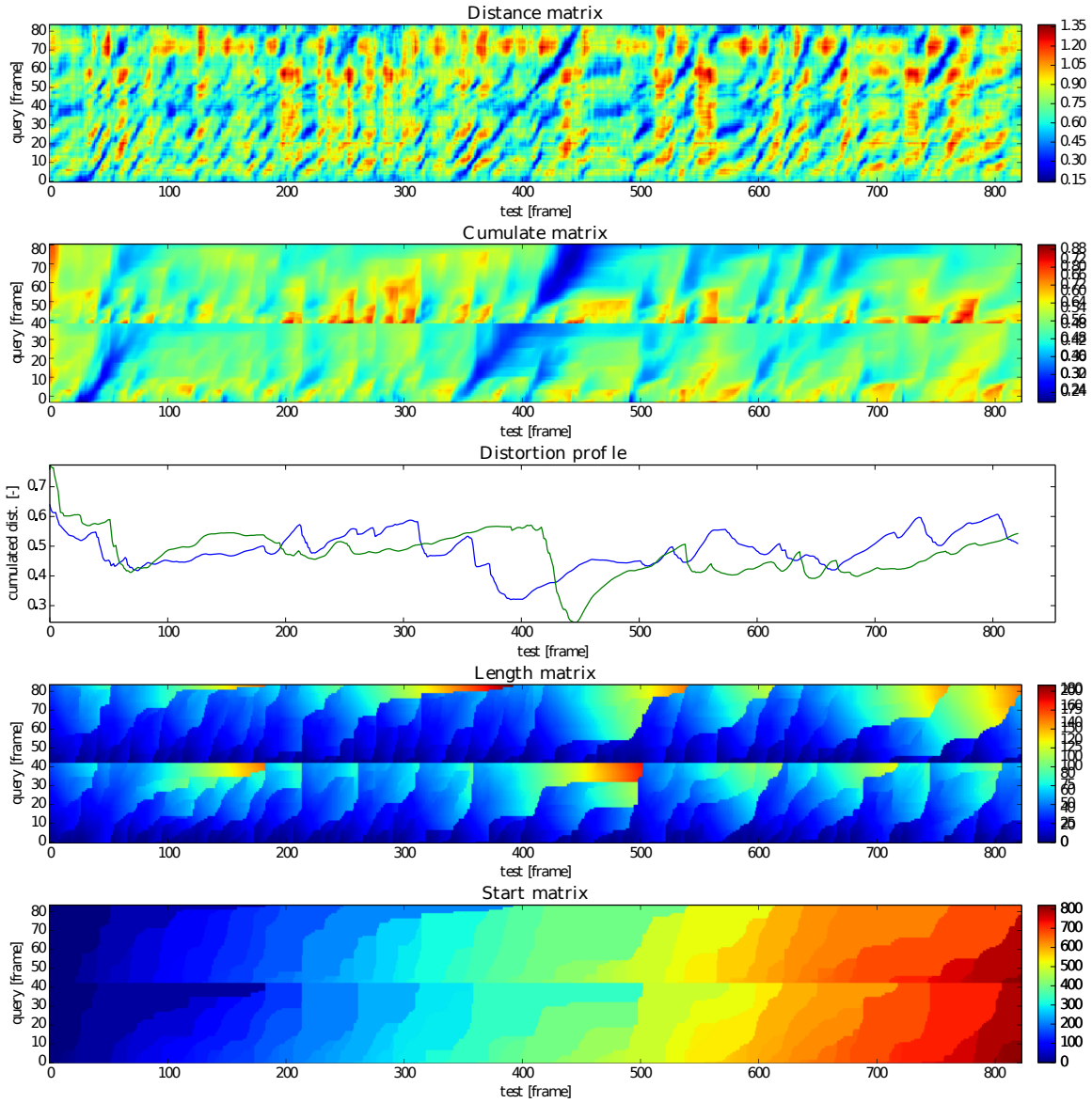


Figure A.1: *Split matrices in half were used to search for Type 3 of query.*