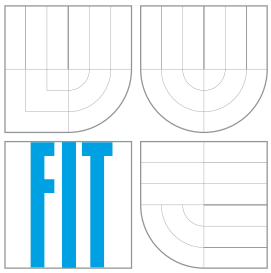
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ VÝRAZŮ V ŘEČI POMOCÍ MLUVENÝCH PŘÍKLADŮ

QUERY-BY-EXAMPLE SPOKEN TERM DETECTION

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE
AUTHOR

Ing. MICHAL FAPŠO

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Dr. Ing. JAN ČERNOCKÝ

BRNO 2014

Abstract

This thesis investigates query-by-example (QbE) spoken term detection (STD), in which queries are entered in their spoken form and searched for in a pool of recorded spoken utterances, providing a list of detections with their scores and timing. We describe, analyze and compare three different approaches to QbE STD, in various language-dependent and language-independent setups with diverse audio conditions, searching for a single and five examples per query. For our experiments we used Czech, Hungarian, English and Levantine data and for each of the languages we trained a 3-state phone posterior estimator. This gave us 16 possible combinations of the evaluation language and the language of the posterior estimator, out of which 4 combinations were language-dependent and 12 were language-independent. All QbE systems were evaluated on the same data and the same features, using the non-pooled Figure-of-Merit metric and our proposed utterrance-normalized non-pooled Figure-of-Merit, which provided us with relevant data for the comparison of these QbE approaches and for gaining a better insight into their behavior. The three presented QbE approaches are: sequential statistical modeling (GMM/HMM), template matching of features (DTW) and matching of phone lattices (WFST). To compare the performance of QbE approaches with the common query-by-text STD systems, for language-dependent setups we also evaluated an acoustic keyword spotting system (AKWS) and a system searching for phone strings in lattices (WFSTlat). The core of this thesis is the development, analysis and improvement of the WFST QbE STD system, which after the improvements, achieved similar performance to the DTW system in language-dependent setups.

Contents

Chapter 1

Introduction

With the growing amount of spoken data, which is recorded, stored and also shared nowadays, the need for its effective indexing and retrieval increases as well.

The most common approach to speech search systems is the "spoken term detection" (STD), which aims at searching a phrase of one or more words in spoken data, outputting a list of detections with score and timing information. Users can enter the search query either in form of text (henceforth, query-by-text or QbT STD[1]) or in form of speech (query-by-example or QbE STD).

Best performing QbT STD systems search in an output of large vocabulary continuous speech recognizers (LVCSR), which are available only for several most widespread languages. Building such recognizer for a new language requires a lot of training data and linguistic resources, so it is not a viable option for most of the world's languages and dialects.

On the other hand, QbE STD systems where queries are entered in their spoken form, can be language-independent and thus they are usually the only option for searching in new or low-resource languages. These systems search in phone posteriorgrams or other, usually unsupervisedly trained, appropriate features. Other applications of QbE STD systems are: searching in multi-language or multi-dialect spoken data, voice-based information systems, searching for out-of-vocabulary words of LVCSR-based QbT STD systems, or for relevance feedback in QbT STD systems.

In this work, we concentrate on the query-by-example spoken term detection. We describe and analyze three different QbE STD systems and compare their performance across several language-dependent and language-independent setups with various audio conditions, using a single or five examples per query. A major part of this work describes our development, implementation and analysis of a QbE STD system which searches in phone lattices or confusion networks derived from phone posteriorgrams.

[1]STD is commonly interpreted as query-by-text STD, but in this work we use the abbreviation QbT STD to differentiate it from generic STD and QbE STD

1.1 Motivation

The main motivation for this work was to gain a better insight into the behavior of most common types of query-by-example STD systems across various language-dependent and language-independent setups with diverse audio conditions. Since there was no such thorough comparison available, we started to work on it. Part of this work has been already published in [Tejedor et al., 2012].

Since we started our effort, MediaEval evaluations targeting language-independent setups of QbE STD were held several times. Participants were also provided with data to evaluate language-dependent setups, but only few participants compared their own systems for both language-dependent and language-independent setups. A global comparison of participating QbE systems was shown in [Metze et al., 2014], but only for language-independent setups. Also, the MediaEval metric is ATWV and MTWV, where results are influenced by calibration of scores across queries, not only by the differences between QbE STD systems themselves. Nevertheless, there was only a single participating QbE STD system searching in lattices [Barnard et al., 2011], where authors tried to use phone lattices to refine search results, but they were unsuccessful in implementing it correctly and didn't report any results with the lattice-based system.

Three main types of QbE STD systems are described in literature, according to the matching technique they deploy: template matching of features (DTW) [Hazen et al., 2009; Szöke et al., 2011; Muscariello et al., 2011], sequential statistical modeling of features (GMM/HMM) [Velivelli et al., 2003; Chan and Lee, 2011; Szöke et al., 2012; Abad et al., 2012] and lattice matching (WFST) [Lin et al., 2008; Parada et al., 2009; Shen et al., 2009; Lin et al., 2009; Barnard et al., 2011]. Out of these, DTW is the most widespread approach. Sequential statistical modeling and lattice matching approaches are used much less.

Actually there were only few published QbE STD systems searching in lattices or confusion networks yet. In [Shen et al., 2009], authors compared several techniques for matching confusion networks of query and utterance, in [Lin et al., 2008, 2009], authors used graphical models for multi-lattice alignment, and in [Parada et al., 2009], authors used transducers to search for out-of-vocabulary words in phone lattices generated by an LVCSR system. In both cases, there was no comparison with other QbE approaches or other languages.

Therefore, we decided to implement the lattice-based QbE STD system and to thoroughly compare all three main types of QbE STD systems. We use a two-blocks architecture in all our STD systems: first block is the feature extractor which encodes the speech into low dimensional feature vectors, and the second block is the query detector (or spoken-term detector) which hypothesizes putative query detections from the features. To make the comparison of all STD systems more relevant, the feature extractor block remains the same in all our STD systems, so the only difference in their performance can be caused by query detector blocks. In this work, we evaluate the QbE STD systems in four language-dependent setups (Czech, English, Hungarian and Levantine), and twelve language-independent setups (all other combinations of the four target languages with four language-specific phone posterior feature extractors), using a single or five examples per query.

1.2 Claims of this Thesis

The goal of this work was to implement a lattice-based query-by-example spoken term detection system and compare it with other state-of-the-art systems across various language-dependent

and language-independent setups. Several chapters of this thesis were partly written by Igor Szőke and František Grézl from the Speech@FIT lab at Brno university and by Javier Tejedor from the HTCLab at Madrid university. We also worked together on several of the presented experiments. My own claims of this thesis are following:

- Analysis of STD evaluation metrics, especially the FOM metric and its proposed unnpFOM variant, which simulates an ideal calibration of scores across utterances.
- Implementation, analysis and improvements of two WFST-based QbE systems. Although these systems were inspired by [Parada et al., 2009], our systems are different and original in several aspects.
- Comparison and analysis of DTW, GMM/HMM and WFST QbE systems in several language-dependent and language-independent setups with a single or five examples per query. I worked on some of the experiments with Igor Szőke from the Speech@FIT lab at our university, and Javier Tejedor from the HTCLab at Madrid university. Our joint work was published in [Tejedor et al., 2012], but many experiments and results are new and original in this thesis.

The source code for our WFST system, scoring tool and other relevant scripts are available at `http://michalfapso.github.io`.

Chapter 2

Experimental Setup

This chapter was originally written by Igor Szőke and František Grézl from the Speech@FIT lab and it was published with few differences in [Tejedor et al., 2012]. Igor and František also trained the feature extractor used for our experiments.

Description of our experimental setup is important for interpretation of our results, for comparing the systems and for reproducibility of our experiments. In this work, we evaluate DTW, GMM/HMM and WFST query-by-example systems on Czech, English, Hungarian and Levantine evaluation datasets. A phone recognizer, trained on different datasets of the same set of languages, produces input data for all the three query-by-example systems.

We used two main setups for our experiments. Language-dependent setup where the phone recognizer was trained on the evaluation data language and language-independent setup where the evaluation language and the language on which the phone recognizer was trained, do not match.

There is an important fact to note here, about our experimental setup, that all our evaluated systems ran on the same data (features, queries, ...) which makes their comparison more relevant. The only exception is the GMM/HMM system which we evaluated, besides the common posterior features, also with bottleneck features.

2.1 Data

In order to inspect the language-independent setup across the different techniques, we trained and evaluated proposed approaches on several languages across different groups: Czech (Slavic), English (Germanic), Hungarian (Uralic), and Levantine (Arabic). We have randomly selected queries with at least five examples in the query training set. In addition, all selected queries are longer than four phones, queries which are substrings of longer queries are discarded and the queries contain only a single word.

2.2 Audio Preprocessing and Feature Extraction

Our QbE STD system is represented in Figure 2.1. It contains four different blocks: voice activity detection (VAD); vocal tract length normalization (VTLN); and speaker mean normalization (SMN) blocks represent standard preprocessing steps. First, a VAD is employed to filter out nonspeech parts of the audio representing the queries and the utterances for proper estimation

	Data (hours)			Queries			Data type
Language	Feature training	Query training	Eval.	Unique	Examples	occurrences	
English	277.7	12.5	2.3	168	840	2007	CTS
Hungarian	8.5	0.9	2.4	8	40	337	PRTS
Levantine	19.9	2.8	2.5	51	255	609	CTS
Czech	100.9	2.2	1.4	58	290	1019	CTS RTS PTS

Table 2.1: Overview of data. The table consists of three parts. First one shows amounts of data used as feature training set, query training set, and evaluation set. The second one shows numbers of queries, examples and query occurrences in evaluation data for each language. And the third part shows the type of data (CTS - continuous telephone speech, PRTS - prompted read telephone speech, RTS - radio telephone speech, PTS - prompted telephone speech.

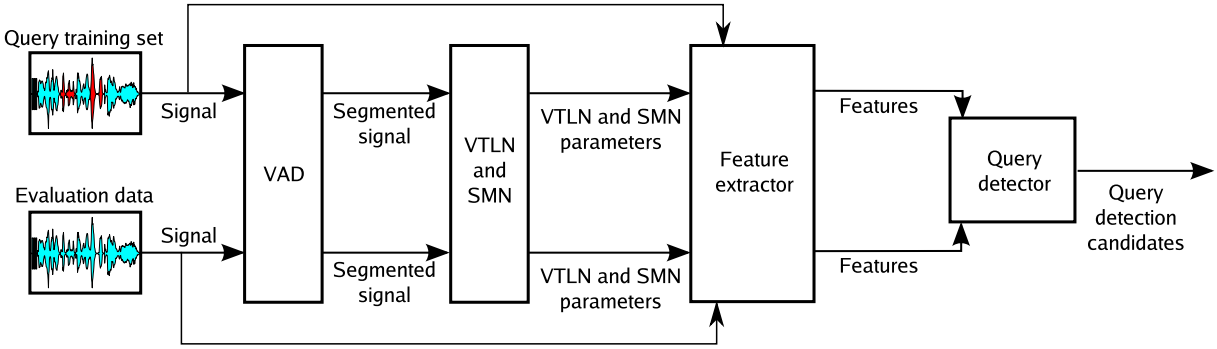


Figure 2.1: High-level schema of our query-by-example STD system.

of the speaker normalization/adaptation parameters in the subsequent step. In the second step, we apply both VTLN and SMN to the remaining audio. The core of this work is the query detector block. All other blocks were developed by the Speech@FIT research group and are briefly described in following sections.

Since one of our goals is to evaluate various QbE STD systems across language-dependent and language-independent setups, the core of our QbE STD systems is split into two blocks: *Feature extractor* encodes the speech in low dimensional feature vectors, and the *query detector* (or spoken-term detector) hypothesizes putative query detections from the features. To localize the differences of our query-by-example systems only into the query detector block, we kept the feature extractor block the same for all our systems.

We experimented with two sets of features: 3-*state phone posteriors* derived from the output of an artificial neural net (NN) classifier and *bottle-neck features*, which are also based on a NN classifier, derived as output of a hidden compression layer of the NN as was described in [Grézl and Fousek, 2008].

2.3 Evaluation

There are several evaluation metrices commonly used in the area of Spoken Term Detection (STD), out of which we chose to use the npFOM metric for its stability, comparability across different datasets and its non-pooled nature which does not require additional calibration of detection scores.

To compute any of these common metrices, an STD system has to provide a list of detections with their start time, end time and confidence score. Then we have to compare detections with a reference transcription and mark each detection either as hit or false alarm.

Let Q be a search query, Δ the set of queries, thr a certain threshold, $N_{target}(Q)$ the number of all occurrences of the query Q in the evaluation data, $N_{nontarget}(Q)$ the number of opportunities for incorrect detection of Q in the evaluation data (constrained by some predefined sampling, e.g. a detection can occur every second), $N_{HIT}(Q, thr)$ the number of detections of the query Q which are identified as hits and their score remains above the threshold thr, and $N_{FA}(Q, thr)$ the number of false detections (i.e., FAs) of the query Q with a score larger than thr. Probability of hit is

$$p_{HIT}(Q, thr) = \frac{N_{HIT}(Q, thr)}{N_{target}(Q)} \tag{2.1}$$

probability of miss is

$$p_{MISS}(Q, thr) = 1 - p_{HIT}(Q, thr) = 1 - \frac{N_{HIT}(Q, thr)}{N_{target}(Q)} \tag{2.2}$$

and probability of false alarm is

$$p_{FA}(Q, thr) = \frac{N_{FA}(Q, thr)}{N_{nontarget}(Q)}. \tag{2.3}$$

In all scoring tools we used, a detection was considered to be a hit in case its start and end times were within a 500ms shift of those of the reference. We chose such a loose time constraints to cancel incorrect timing errors possibly introduced by converting posteriors to lattices and further to confusion networks. When more detections belong to the same reference, we take only the one with the highest score among them and remove the others from scoring.

2.3.1 ROC and FOM

Receiver operating characteristics (ROC) for a particular query Q is a $p_{HIT}(Q, thr)$ as a function of $N_{FA}(Q, thr)$ per hour (see Figure 2.2 for an example). The larger the area under the ROC curve is, the better the system performs.

Figure of merit is a keyword-spotting accuracy averaged over 1 to 10 false alarms per hour. The FOM calculation assumes that the total duration of the evaluation speech is T hours. All detections are sorted by score and a hit percentage $P_{HIT}(i)$ of queries found before the ith false alarm is calculated for $i = 1 \ \ldots \ N+1$ where N is the first integer $\geq \ 10T - 0.5$. Figure of merit is then defined as

$$FOM \ = \ \frac{1}{10T}(P_{HIT}(1) + P_{HIT}(2) + \ldots + P_{HIT}(N) + a\, P_{HIT}(N+1)), \tag{2.4}$$

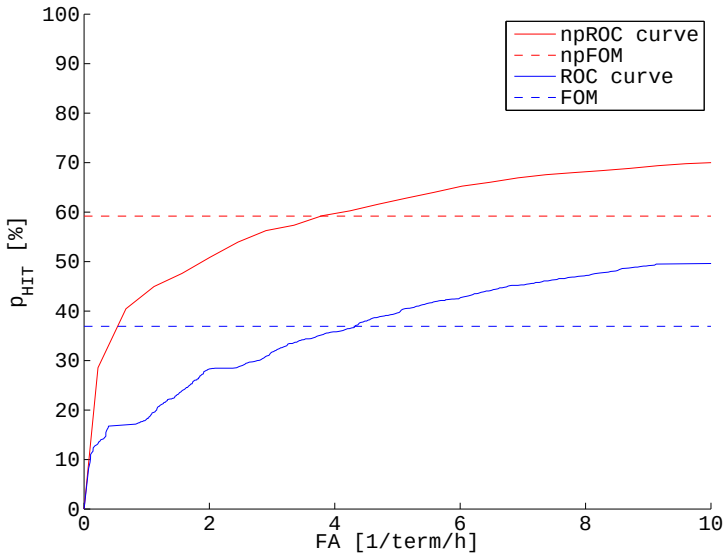


Figure 2.2: Example of a receiver operating characteristics (ROC) and figure of merit (FOM), showing pooled and non-pooled variants of the metrics. The huge difference between pooled and non-pooled variants in this particular example is caused by an inconsistency of scores for different queries. If scores across all queries were well calibrated, pooled results would be much closer to non-pooled results.

where $a = 10T - N$ is a factor that interpolates to 10 false alarms per hour and $P_{HIT}(i)$ is:

$$P_{HIT}(i) = \sum_{Q \in \Delta} \frac{N_{HIT}(Q, thrFA(i))}{N_{target}(Q)} \times 100\%, \tag{2.5}$$

where the auxiliary function $thrFA(i)$ finds the proper threshold thr for the ith false alarm in all queries. Since this metric does not distinguish individual queries, the $thrFA(i)$ threshold should be ideally similar for all evaluation queries and thus it requires a proper calibration of detection scores across different queries.

According to the FOM definition in [Young et al., 2006], a putative detection is considered to be a hit in case the midpoint of a reference occurrence is between the start and end times of the given detection. However, in our implementation we loosen the time constraints to cancel the effect of incorrect timing in lattices and confusion networks and we stretch each reference by 0.5s to both directions and when more detections overlap with the same reference, we keep only the one with the highest score and the others are discarded from scoring.

Higher value of FOM means better performance. A drawback of the FOM metric might be the artificial restriction of false alarms per hour from 1 to 10. Even though there are applications which could operate even with a higher number of false alarms, this restriction is valid for most real use cases.

For evaluating our experiments, we used a non-pooled variant of the FOM metric which calculates the threshold in (2.5) for each query independently and it is therefore more suitable for evaluating STD systems in early stages of their development, as it does not require calibration of scores across queries. We also used another non-pooled FOM metric which simulates an ideal

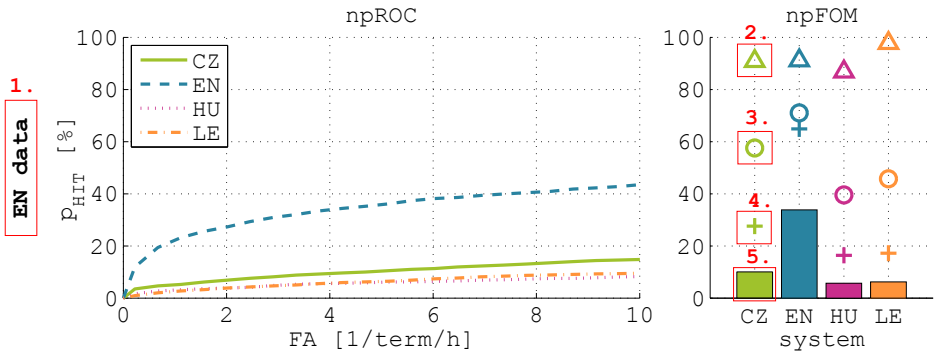


Figure 2.3: Example of results figure as used for presenting results in this work. Left part contains npROC curves, right part npFOM metric and its derivatives, unnpFOM and oracleFOM. "1." denotes the evaluation dataset, either one of the four languages (CZ, EN, HU, LE) or a language dependent setup, where the dataset language corresponds to each system. "2." shows the oracleFOM, "3." unnpFOM, "4." is the maximum npFOM of all 5 examples of queries and "5." is npFOM.

calibration across evaluation utterances, denoted in this work as *unppFOM*, the "utterance-normalized npFOM", where the threshold in (2.5) is optimized for each query and each utterance independently. We used yet another metric, the oracleFOM, which simulates ideal ordering of hits and false alarms and shows whether or not a reference occurrence was detected, irrespective of detection scores.

2.3.2 Reading Result Figures

Throughout this work, we present results in form of figures which we find more readable than tables. An example with description of important figure elements is shown in Figure 2.3. For comparing multiple results, we omit npROC curves and show only npFOM part of these figures.

Chapter 3

Contrastive QbE STD Systems

Although the core of this thesis is the WFST-based QbE STD system, we have to compare its performance with other QbE systems to make conclusions. In this chapter, we briefly describe them.

3.1 Acoustic Keyword Spotting (AKWS) – Upper-bound Experiment

The acoustic keyword spotting system's setup and experiments were done by Igor Szőke from the Speech@FIT lab, where the AKWS system was developed.

We took the acoustic keyword spotting as our upper-bound technique and also derived the GMM/HMM approach for QbE STD from it. A full description of the acoustic keyword-spotting system can be found in [Szöke, 2010].

3.2 GMM/HMM-based Query-by-Example Detector

The GMM/HMM QbE system setup was done by Igor Szőke from the Speech@FIT lab. This system has similar architecture as the AKWS system, only the query model is not derived from a pronunciation dictionary, but it is trained from query examples. In our experiments, we set the number of HMM states for each query according to its real number of phones looked up in the pronunciation dictionary. The query model is then trained on query examples. Background model in this system is represented by a GMM trained on a whole query training set. Similarly to the AKWS system, the background model and query model are both matched against utterance features, providing a likelihood ratio for each frame, which represents score of a detection ending in that particular frame. Besides 3-state posterior features used for all other systems, for the GMM/HMM we experimented also with bottleneck features which showed better performance.

3.3 DTW-based Query-by-Example detector

The DTW-based system was developed by Miroslav Skácel from the Speech@FIT lab, the system for combining query examples was developed by Javier Tejedor from the HTCLab at Madrid university, who also helped us with experiments.

Our DTW-based QbE detector relies on template matching of 3-state phone posterior features, computing a distance matrix between the query and utterance and finding best paths traversing the whole query and any part of the utterance. For a similarity function, we used log-likelihood based on the cosine distance. For measuring paths through the distance matrix, we used cost 1 for right or down steps and $\sqrt{2}$ for diagonal steps.

To combine five examples of a query into one posteriorgram, we used the method described in [Tejedor et al., 2010], where we first matched all examples with each other and sorted them by their similarity. Then we took the best example and mixed features of all other examples into it by following the warping paths, with weight $\frac{1}{8}$ for each of the four examples.

Chapter 4

WFST-based QbE detector

The WFST-based QbE detector forms the core of this thesis. Description of the WFSTlat system working with lattices was published in [Tejedor et al., 2012]. In this thesis, we developed the system further and achieved substantial improvements.

All the approaches presented earlier used phone posterior features directly when searching for queries. In this section, we work with phone lattices or confusion networks, derived from phone posterior features, representing both queries and utterances. This approach aims at finding all occurrences of a query lattice inside an utterance lattice, while preserving the timing and score of each occurrence. Weighted finite state transducers (WFSTs) offer a well-defined framework for this purpose.

Inspired by the work of [Parada et al., 2009] who aimed at searching OOV terms using WFSTs, we focus on the language-independent aspect of QbEand instead of using LVCSR or hybrid word/phone lattices, we create phone lattices from posteriors generated by the phone recognizer described in Section 2.2.

4.1 Analysis of the WFSTlat system

Results comparison of the WFSTlat system presented in [Tejedor et al., 2012] and DTW-based QbE system for language-dependent setups is shown in Figure 4.1. We can see that DTW systems significantly outperform WFSTlat systems for both single and five examples per query. The performance gap for the single example case is the main issue of the WFSTlat system. The case of five combined examples is influenced also by different combination strategies used for the two QbE systems, since the DTW system combines posteriorgrams of five examples to create a single posteriorgram representing the query, and WFSTlat system searches for each of the five examples and combines the detections afterwards.

In this section, we analyze possible causes of the performance gap between the DTW and WFSTlat systems.

4.1.1 From Posteriorgrams to Lattices and Back

During the conversion from phone-state posteriorgrams to lattices, certain amount of information is lost. In Section **??** we already compared the performance of the AKWS and WFSTdict systems, where the former one works with full posteriorgrams and the second one with phone lattices, but queries for these systems were simple phone strings taken from a pronunciation

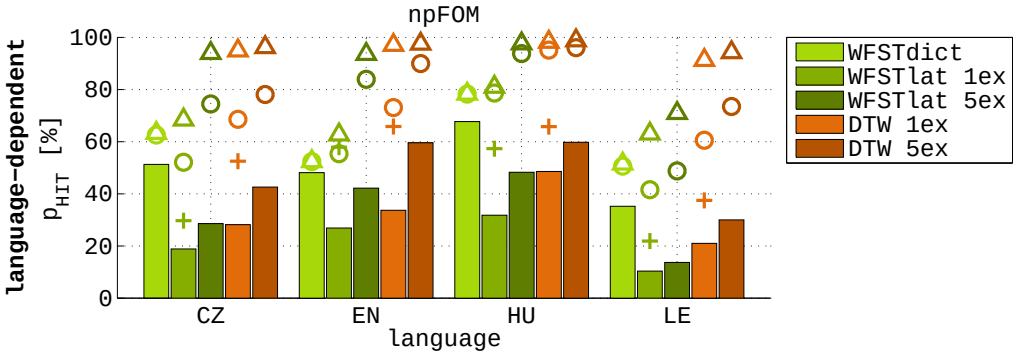


Figure 4.1: Results comparison of query-by-text WFSTdict systems and WFSTlat QbE STD systems with single and five examples per query for the language-dependent setups. The systems search in lattices with 700 links per second.

dictionary. In this section, we will analyze the difference between full posteriorgrams and lattices from the perspective of a query-by-example system. We used DTW search over original posteriorgrams and over posteriorgrams generated from lattices to analyze the amount of information loss introduced by converting posteriorgrams to lattices. This way we can isolate the cause of potential performance degradation only to the information loss introduced by lattices themselves, since the WFSTlat system is not present at all in this experiment. The original full phone-state posteriorgrams (Figure 4.3a) were considered a baseline for this experiment.

4.1.1.1 Posteriorgrams to Lattices

Phone-state posteriorgrams were converted to phone-state lattices using the HVite decoder. We experimented with two types of decoding networks. One where the order of phone states is not constrained, so that any phone state can be active in any frame. And another one where the order of phone states is constrained, so that after the first state of a phone, only the second state of the same phone can follow and switching to another phone is possible only from the last state of the phone.

4.1.1.2 Lattices to Posteriorgrams

Phone-state lattices were used for pruning the original posteriorgrams. We traversed lattices and for each link we copied the corresponding posteriors from the original posteriorgram to a new pruned posteriorgram. This way we created posteriorgrams pruned by lattices. Examples of posteriorgrams generated from an unconstrained and constrained phone-state lattices are shown in Figures 4.3b and 4.3c respectively.

Another set of posteriorgrams was created by computing each link's posterior probability in lattices and adding the probability to a new posteriorgram on the corresponding position. In this case we didn't use any direct information from the original posteriorgrams, those new posteriorgrams were generated directly from lattices.

Posteriors with zero probability were raised to slightly above zero to better fit the distance metric of the DTW system.

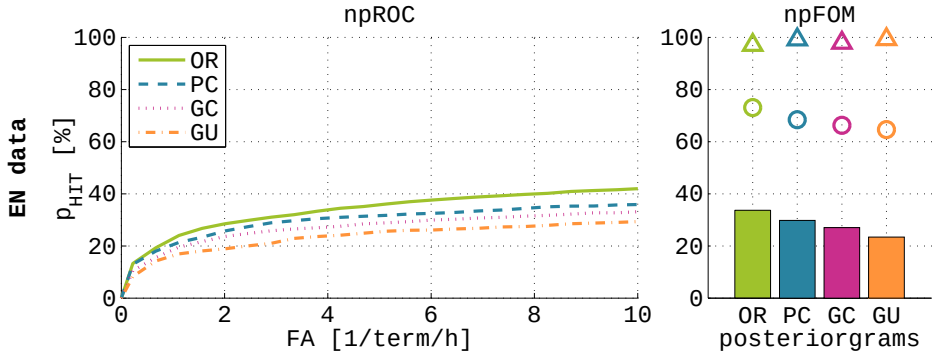


Figure 4.2: Results of DTW-based QbE system with language-dependent setup for English searching for a single example per query in four different versions of posteriorgrams: original full posterior features (OR), posteriorgrams pruned by constrained phone-state lattices (PC), posteriorgrams generated from constrained phone-state lattices (GC), posteriorgrams generated from unconstrained phone-state lattices (GU).

4.1.1.3 Results

We run this experiment only for the English language-dependent setup. Results in Figure 4.2 show that the performance drops for posteriorgrams pruned by constrained phone-state lattices, it drops further when posteriorgrams are directly generated from constrained phone-state lattices, and even further when posteriorgrams are generated from unconstrained phone-state lattices. Thus the loss of information introduced by converting posteriorgrams to lattices is not negligible and we can not expect same results for WFST QbE system as for the DTW QbE system which works with full posteriorgrams.

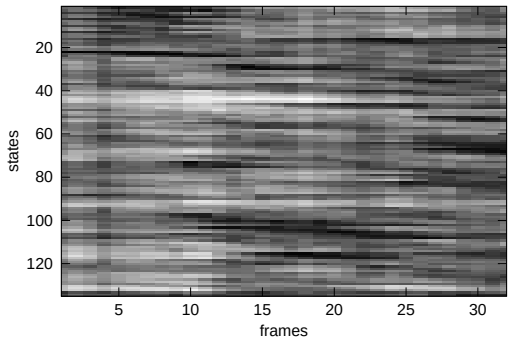
4.2 Improvements

In this section, we describe several improvements over the basic WFSTlat system published in [Tejedor et al., 2012].

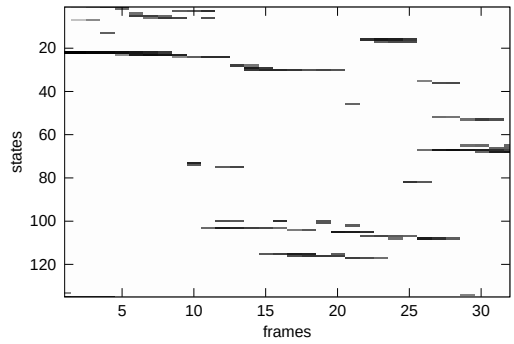
4.2.1 Confusion Networks

Because the WFSTlat system working with lattices performed considerably worse than the DTW system working with full posteriorgrams, we decided to develop and experiment with a confusion network-based WFST system.

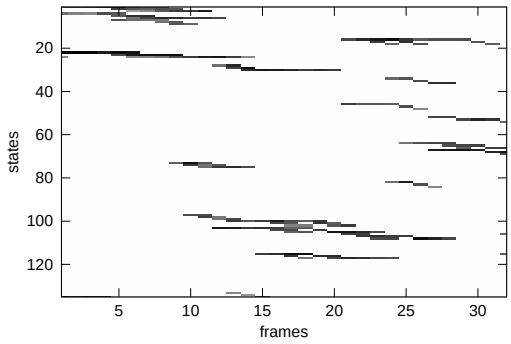
Phone lattices described in [Tejedor et al., 2012] can be further converted to confusion networks. They have a special property that each path from the start node to the end node goes through all other nodes, while all paths from the original lattice are preserved and some new paths are added. Confusion networks are more efficient than traditional lattices, in terms of size and structure, without compromising recognition accuracy. Since they force the competing phones to be in the same group, they enforce the alignment of the words that occur at the same approximate time interval in the lattice. Phones in confusion networks have posterior probabilities, which are basically the sum of the probabilities of all paths which contain that phone at



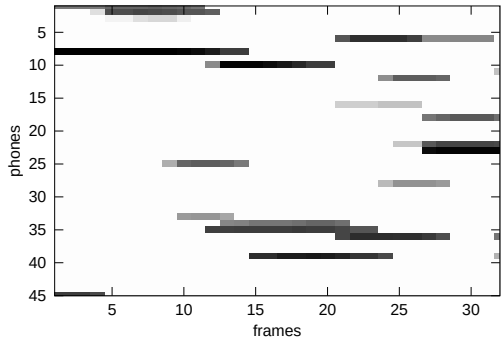
(a) Original posteriorgram.



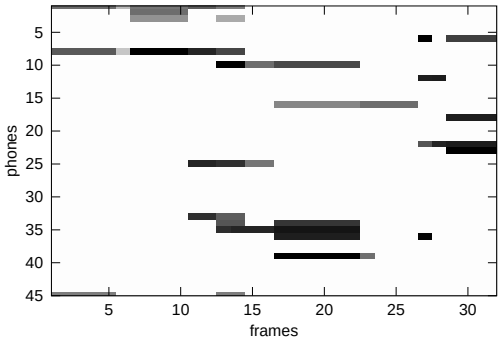
(b) Posteriorgram generated from an unconstrained state lattice.



(c) Posteriorgram generated from a constrained state lattice.



(d) Posteriorgram generated from a phone lattice.



(e) Posteriorgram generated from a phone confusion network.

Figure 4.3: Comparison of posteriorgrams of one example of the term "ACTUALLY". Pruning factor for generating lattices was chosen to emit approximately 700 links per second. These figures visualize the information loss during conversion of posteriorgrams to lattices and further to confusion networks.

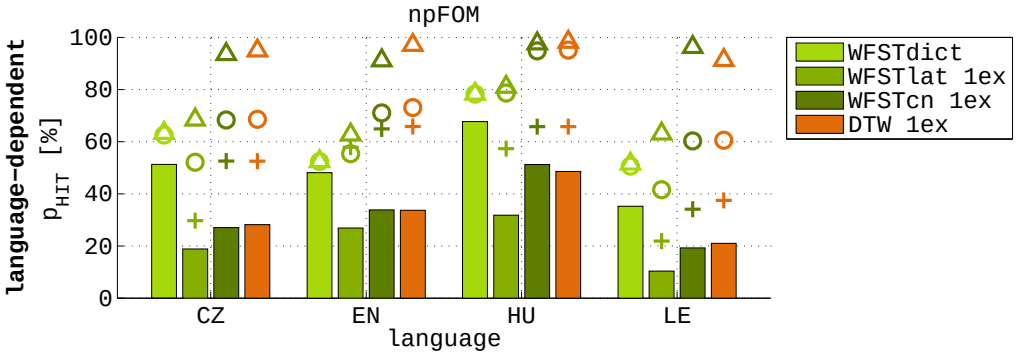


Figure 4.4: Results comparison of WFSTlat and WFSTcn systems with language-dependent setup searching for a single example per query. WFSTdict

around that approximate time frame and for each group they sum to one [Mangu et al., 2000; Hakkani-Tür et al., 2006].

The process of converting confusion networks to transducers is similar to the one for lattices described in [Tejedor et al., 2012]

4.2.1.1 Single Example per Query

We first consider the basic language-dependent setup where we search only for a single example per query. The results for the confusion network-based WFST QbE system (WFSTcn) together with the WFSTdict, WFSTlat and DTW systems for comparison, is shown in Figure 4.4. We can see in the figure, that the WFSTcn system significantly outperformed the WFSTlat system, and is now on par with the DTW system for Czech and English, slightly better for Hungarian and slightly worse for Levantine. Also the unnpFOM metric is now very similar for both WFSTcn and DTW systems.

The reason why using confusion networks instead of lattices caused such a significant increase in performance, could be in the algorithm for merging parallel detections, where the WFSTlat system implements a more simple algorithm than the SRILM lattice-tool used for converting lattices to confusion networks. Also for lattices, the number of parallel links in the R transducer is much higher, which makes the technique for combining overlapping parallel paths much more important than in the case of confusion networks in the WFSTcn system. Another important difference is that probabilities of overlapping links belonging to the same time-aligned group are summed up, while in WFSTlat system, for each group of overlapping detections, only a single best detection is taken. As we wrote before, the technique used for merging overlapping detections, seems to be an important factor influencing the WFST QbE system's performance. In addition to these reasons, confusion networks were reported to outperform lattices also in query-by-text STD [Mangu et al., 2014].

Let us now analyze the results for language-independent setups, still with only a single example per query. The comparison of WFSTlat, WFSTcn and DTW systems is shown in Figure 4.5, where we can see that for language-independent setups, the WFSTcn system still perform considerably worse than the DTW system. However, we should also note that for the best example case ("+" marker in the figure), for Czech data with Hungarian and Levantine feature extractors, the WFSTcn system performs even better than the DTW system, although

the unnpFOM values are still better for the DTW system. Investigating the unnpFOM metric across the whole figure, we can see that the difference between WFSTcn and DTW systems is relatively low, which means that the WFSTcn system is worse calibrated across different evaluation utterances than the DTW system.

4.2.1.2 Combining Examples

Several strategies are appropriate for combining multiple examples per query. In our initial experiments with the WFSTlat system, we used the post-search "merge detections" strategy, where probabilities of overlapping detections were summed up and the timing was taken from the best detection among them. For the WFSTcn system, we experimented also with taking only the single best detection of each overlapping group. Since the DTW system uses the "merge posteriors" strategy, we tried that one also for the WFSTcn system, where we generated query lattices from the merged posteriorgrams. Results for language-dependent setups are shown in Figure 4.6. They reveal that the posteriors merging strategy performs significantly better than the post-search merging of detections.

Both post-search merging strategies seem to perform similarly, only for Levantine summing up probabilities of overlapping detections seem to outperform the case of taking only a single best detection of the group. It means that for Levantine, more vividly then for the other languages, the more examples agree on the same detection, the better score the detection should have, while also taking into account the actual scores of overlapping detections.

Another important observation revealed in the figure is, that the performance of WFSTlat and WFSTcn systems for the post-search merge detections strategy is similar, which does not correspond to their performance for the single example per query case (Figure 4.4). For Hungarian, the performance of WFSTlat system is even considerably better than that of the WFSTcn system. So the merging strategy seems to hurt the performance the more, the better the system performs for individual examples. However, the reason for this behavior could be of course more complex and has yet to be investigated more deeply.

Also when we inspect the unnpFOM metric for Czech and Hungarian systems, we see that the post-search merge of detections and the merge of posteriorgrams perform similarly, which means that the main cause of their significant difference in npFOM values is caused by a better calibration of merged posteriorgrams across different utterances.

Overall results comparison for both single and five examples per query cases is shown in Figure 4.5, where five examples for the WFSTlat system are combined by the post-search merge strategy and for the WFSTcn system by merging posteriors. We can see that also for language-independent setups, merging of posteriors outperforms the post-search merging of detections. Although for language-dependent setups with five examples per query, the performance of WFSTcn system is similar to that of the DTW system, for language-independent setups, WFSTcn performs worse. The possible cause of this performance deterioration seems to be again in the technique of merging overlapping detections in the R transducer. In language-independent setups, the number of overlapping detections in the R transducer is higher than in language-dependent setups. This could lead to a larger difference between considering only a single best detection out of an overlapping group of detections, and taking into account all detections in the group. As we wrote before, we tried summing up probabilities of overlapping detections, but results were significantly worse than for taking only a best detection.

Also we should note, that the post-search example merging technique, either taking a best

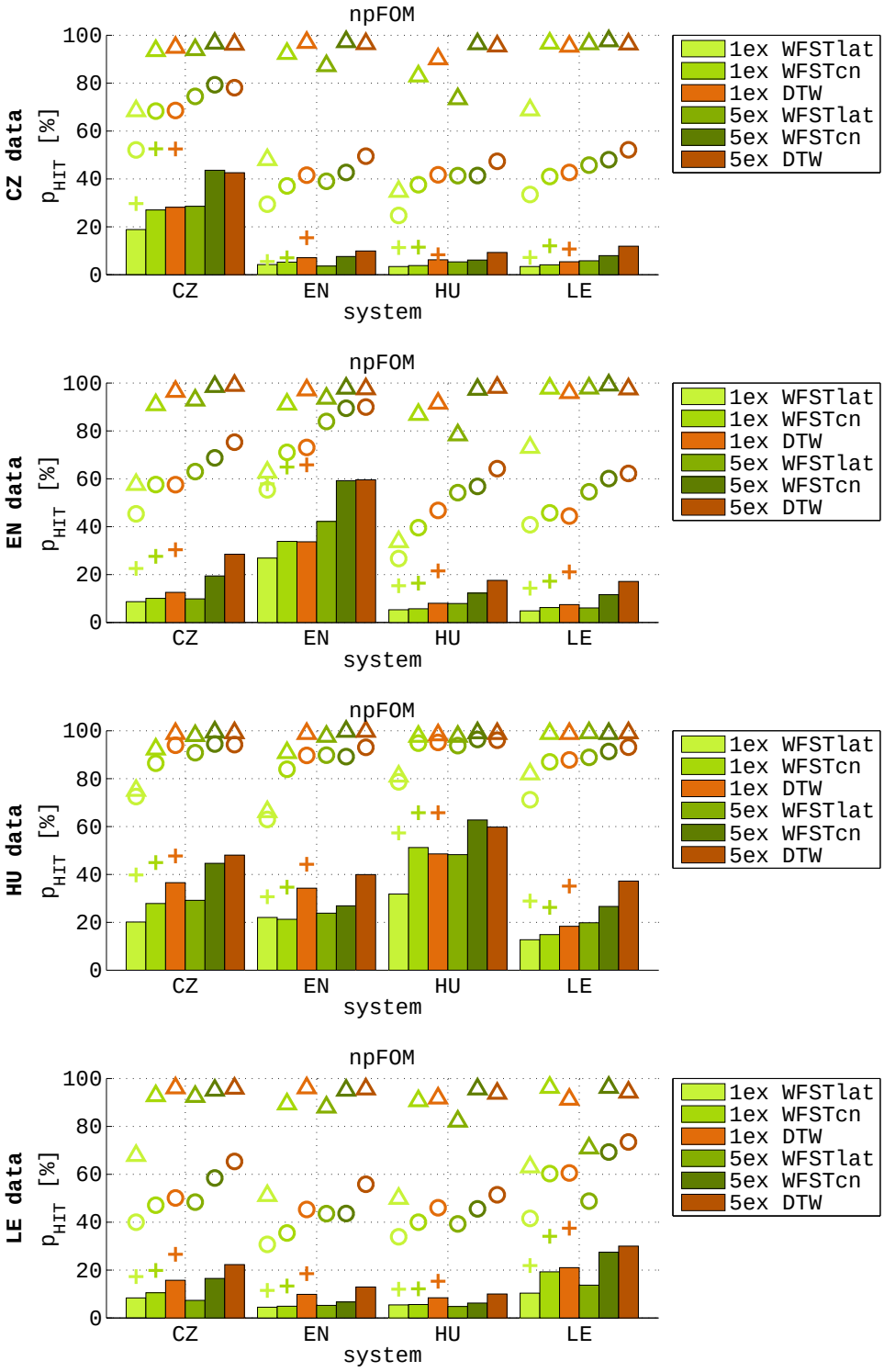


Figure 4.5: Results comparison of WFSTlat, WFSTcn and DTW QbE STD systems searching for a single or five examples per query in both language-dependent and language-independent setups.

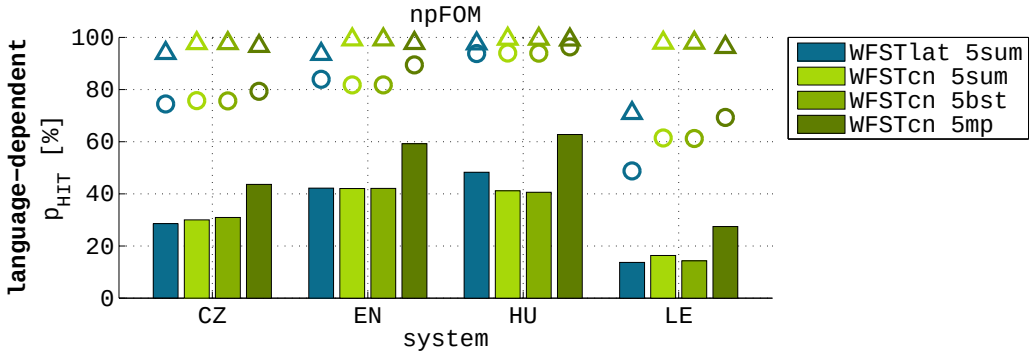


Figure 4.6: Results comparison of various strategies for combining five examples per query in WFSTlat and WFSTcn systems and language-dependent setups. "5sum" stands for the post-search "merge detections" strategy, where scores of overlapping detections are summed up (*logadd* in our case of log scores), "5bst" selects only the best score among the overlapping detections, and "5mp" stands for the "merge posteriors" strategy where for each query, posteriorgrams of all five examples are first merged into a single posteriorgram, and then the QbE system works as in the single example per query case.

path or summing up weights of overlapping paths, is basically the same merging technique as is used in the process of extracting detections from the R transducer, when they overlap in time. As we have seen in Figure 4.6, this merging technique is suboptimal, inferior to merging of posteriorgrams. When the same suboptimal technique is used in the process of extracting detections from the R transducer, we get suboptimal results there as well. As we have pointed out already, there are more overlapping detections in language-independent setups, which is probably the reason for their worse performance.

4.2.2 Dealing with Silence

Silence in speech data is handled in our phone recognizers by a *sil* phone model, which is then propagated from posteriorgrams to lattices and confusion networks. It may happen that in examples cut from audio data, silence appears at the beginning or end of an example. Then the silence is expected to be contained also in detections of that example in evaluation utterances. If time boundaries of our query examples were perfectly correct, there would be no silence at the beginning and end of the examples. We even know apriori that examples of our queries are all single words, so they should not contain any silence at all.

Thus we tried to set zero probability for silence in query posteriorgrams, which effectively discarded silence also from lattices and confusion networks. Results of these examples without silence (further denoted by "NOSIL") were very similar to the original examples ("ORIG"). For language-dependent setups, npFOM improved by 0.66% absolute in average, for language-independent setups by 0.2%. However, when we take best results of both NOSIL and ORIG setups for each query, npFOM improves by another 0.46% for language-dependent and by 0.37% for language-independent setups. It means that some query examples perform better with silence discarded from them, while other examples perform better with silence untouched.

Since the overall results are better for examples without silence, in all our experiments with the WFSTcn system, silence was discarded from all query examples.

Chapter 5

Overall Results and Discussion

In this section we compare and discuss results of our experiments with systems described earlier in Chapters 3.1, 3.3, 3.2 and 4. This chapter was previously written with Igor Szőke from the Speech@FIT research group at Brno university and Javier Tejedor from the HTCLab at Madrid university, and was published in [Tejedor et al., 2012], but it was completely rewritten in this thesis.

5.1 Language-Dependent

Let us first compare all systems for language-dependent setups in Figure 5.1. For the single example per query case, GMM/HMM system performs the worst, needing more than a single example of a query to reliably train its model. It is outperformed by DTW and WFSTcn systems which achieve very similar overall performance (npFOM, unnpFOM and oracleFOM) for all languages and clearly outperform the GMM/HMM in the single example per query case. Although their npFOM values are still far from the AKWS and WFSTdict baselines, when we artificially select a single best example out of the five examples for each query ("+" markers in the figure), performance gets on par with baselines. For English, the best example case even outperforms the WFSTdict baseline which is probably caused by insufficient density of English lattices when a query is represented by only a single phone string, or it might be also caused by slightly different pronunciation in the pronunciation dictionary than in actual evaluation occurrences.

By inspecting the systems with five examples per query, we can see that the performance of all systems increased significantly, however for Hungarian the increase was lowest among all four languages. This is probably caused by the read prompted speech audio condition in Hungarian data, which results in high similarity of the five examples and their occurrences in evaluation data for each query.

With five examples per query, the GMM/HMM system improved significantly, especially for English and Levantine, where it reached the performance of DTW and WFSTcn systems, for Levantine it even slightly outperformed them. However, for Czech and Hungarian it again performed worse. DTW and WFSTcn systems perform consistently and similarly well for all four languages. Comparing npFOM of the AKWS baseline and the best of the three QbE systems for each language, the deterioration is around 14% in average (20% for CZ data with WFSTcn, 8% for EN data with DTW, 8% for HU data with WFSTcn and 22% for LE data

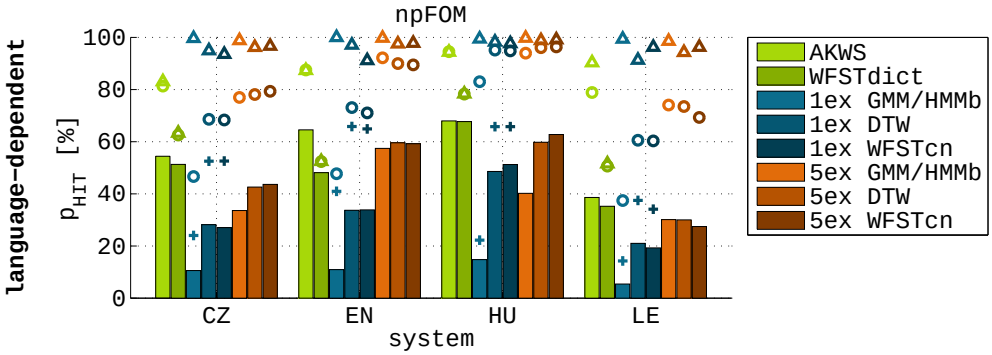


Figure 5.1: Results comparison of QbE systems in language-dependent setups. The "b" suffix of GMM/HMM system denotes bottleneck features. All other systems use posterior features or confusion networks derived from them. AKWS and WFSTdict baseline systems are in green, GMM/HMM, DTW and WFSTcn for single example per query are in blue and for five examples per query in brown.

with GMM/HMM).

We can conclude that for language-dependent setups, DTW and WFSTcn QbE systems perform better than the GMM/HMM QbE system. The performance of an artificially selected single best example for DTW and WFSTcn systems shows, that the technique for merging examples is still suboptimal and can be potentially improved, which could lead to performance similar to the baseline AKWS system.

5.2 Language-Independent

Overall results for all language-dependent and language-independent setups are shown in Figure 5.2, where we also show the GMM/HMM system working with phone posteriors, for clearer comparability of the three QbE STD approaches working with exactly the same set of features – 3-state phone posteriors. In this section we focus only on language-independent setups.

For the case of a single example per query, DTW performs the best of the three QbE systems in all language-independent setups. GMM/HMM and WFSTcn systems perform similarly in most cases, only for Hungarian data with Czech feature extractor, WFSTcn performs considerably better than the GMM/HMM system.

Let us now analyze the case of five examples per query in language-independent setups for each system separately.

The GMM/HMMb approach is the most accurate of the three QbE systems, having best results with the Czech feature extractor. This is caused by large amount of data used to train the feature extractor and also by the mixed audio conditions of Czech data, which makes the system more robust. English feature extractor performs also very well with this system, but although its amount of training data was large, they contained only the CTS audio condition which probably made the system less robust for language-independent setups then the Czech system. We can see, that considering only the GMM/HMMb system, a language-independent feature extractor may achieve comparable results to those obtained with a language-dependent GMM/HMMb system in case of more challenging conditions, e.g. nondiactitized data in Levantine or low amount of

training data for Hungarian. Also for the GMM/HMM system, bottleneck features were found to effectively compress the important information contained in 3-state phone posteriors. Together with their Gaussian-like distribution, it lead to better trained query models in the GMM/HMM system with five examples per query.

The DTW-based QbE STD system achieves about 50% to 75% of precision of the GMM/HMMb system in language-independent setups, except for Hungarian data, where the DTW system gets closer to the GMM/HMMb. This is caused by the small amount of training data used to train the GMM/HMMb query background model. It confirms our conjecture that a model-based approach is able to deal with the phone posterior uncertainty in a language-independent setup where enough training data is available.

The WFST approach shows the lowest performance of the three QbE systems for language-independent setups, where its performance is even lower than that of the DTW system, similarly to the single example per query case. This is most probably caused by a suboptimal method for merging overlapping detections.

5.3 Combining Systems

Results of a very naive combination of the three QbE systems, where for each query, we select the best result among a set of systems, is shown in Figure 5.3. It shows, that especially for language-dependent setups, all three systems are complementary to some extent, so that there are some queries having best results with one particular QbE system. For language-independent setups, the perfomance gain of this simple way of combining systems is less significant and it is sufficient to combine GMM/HMM system with DTW. Combination of GMM/HMM and WFSTcn systems performs slightly worse for language-independent setups.

According to these preliminary results, it should be possible to fuse the three QbE systems to increase the performance.

5.4 Practical Considerations

In real world applications, the system accuracy may not be the only criterion. We should also take into account the speed of indexing and search and the amount of disk space needed to store the utterances.

The phase of extracting features is the same for all our QbE systems. The GMM/HMM system then has to train or adapt the background model. The WFSTcn system has to convert phone-state posteriorgrams to lattices and then further to confusion networks. However, these steps are relatively fast for all the systems. Where the real difference comes, is in the search speed. However, we have to also note here, that our implementations of the three QbE systems were not specifically optimized for speed.

There are techniques for speeding up the DTW search [Zhang and Glass, 2011; Schmalenstroeer et al., 2011; Mantena and Anguera, 2013; Anguera, 2013], but in its basic implementation, it is certainly the slowest of the three systems. GMM/HMM and WFSTcn systems are, in our implementation, approximately on par in terms of search speed. However, an efficient inverted index can be created for the WFSTcn system to significantly speedup search times [Can and Saraclar, 2011].

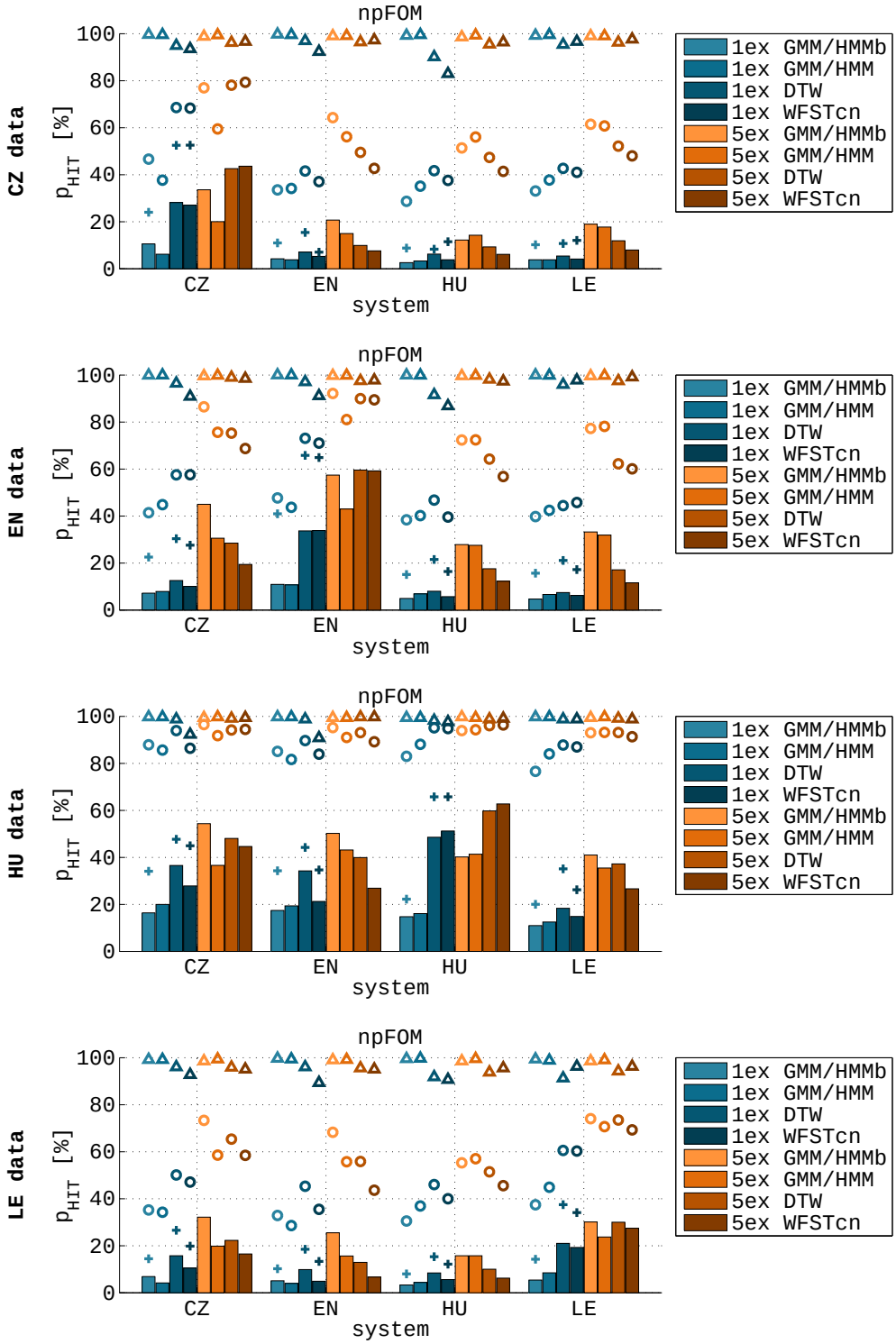


Figure 5.2: Results comparison of the GMM/HMM, DTW and WFSTcn QbE STD systems in both language-dependent and language-independent setups. The "b" suffix of GMM/HMM system denotes bottleneck features. All other systems use posterior features or confusion networks derived from them. Single example per query cases are in blue and five examples per query in brown.

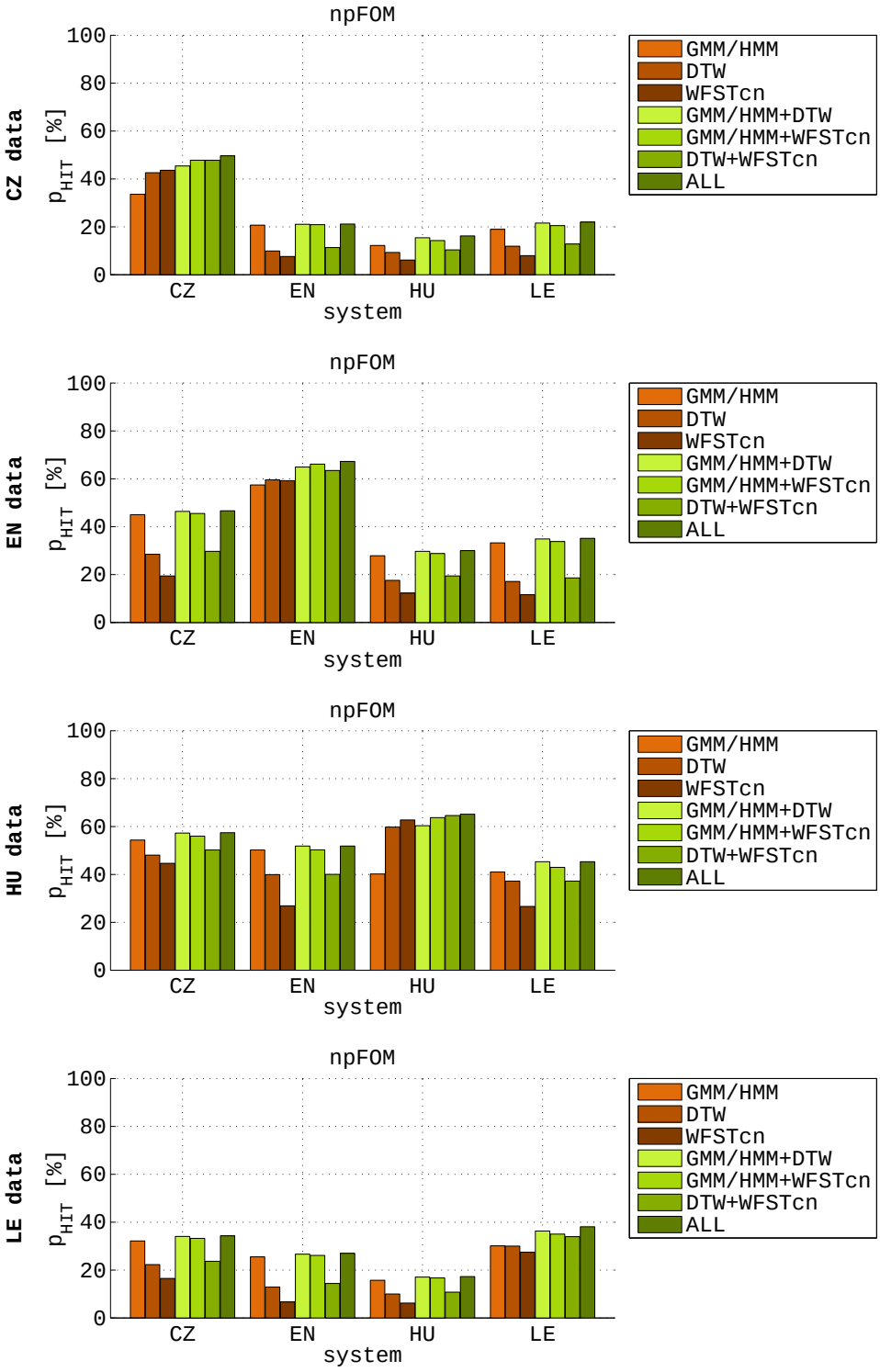


Figure 5.3: Results comparison of the GMM/HMM, DTW and WFSTcn QbE STD systems in both language-dependent and language-independent setups. Single example per query cases are in blue and five examples per query in brown.

Considering the disk space, DTW system has highest requirements, because 3-state phone posteriorgrams are large themselves. The GMM/HMM system works with bottleneck features which require around 20% of the disk space needed for posteriorgrams. Finite state transducers for the WFSTcn system require even less: 3% of the disk space needed for posteriorgrams. Moreover, the size of transducers can be tuned for a particular application.

Chapter 6

Conclusions and Future Work

In this work, we have presented three query-by-example STD systems and two query-by-text STD baselines. All systems were described, evaluated and analyzed in language-dependent and language-independent setups, for a single and five examples per query. In our experiments, all STD systems worked with the same set of features, while GMM/HMM worked, besides the common 3-state phone posterior features, also with bottleneck features.

We found out, that for language-dependent setups, for both a single and five examples per query, DTW and WFSTcn QbE STD systems achieve best performance. For five examples per query, performance of all three QbE systems increases significantly, although it is still not as good as that of the baseline AKWS system. For clean Hungarian data, WFSTcn slightly outperformed DTW, while for more challenging nondiacritized Levantine data, DTW performed slightly better. For Levantine and English data with five examples per query, also the GMM/HMM bottleneck system caught up the other two systems, but for Czech and Hungarian its performance is still considerably inferior to the other two systems.

For language-independent setups with a single example per query, the DTW system outperforms the other two systems, but with five examples per query, the GMM/HMM bottleneck system improves significantly and outperforms the others. Our experiments also showed that bottleneck features are more sensitive to amount of training data than 3-state phone posterior features.

Based on the presented results, we can conclude that query-by-example STD systems are a viable alternative to query-by-text STD systems, especially when more examples per query are available.

As we expected, language-independent setups show a significant decrease of performance, compared to language-dependent setups. However, there is an exception to this behavior, seen in Levantine data, where the Czech GMM/HMM bottleneck system outperformed even all language-dependent Levantine QbE systems. In general, we can say that the mismatch between training and target languages of a feature extractor, are the main reason for the significant decrease of performance in language-independent setups.

In this work, we have also analyzed and significantly improved our WFST system, which now performs on par with the DTW system in language-dependent setups. Moreover, the WFST system's search is considerably faster and requires only about 3% of the disk space, compared to the DTW system.

6.1 Future Work

During our work, we saw many possible directions for future research, which we were not able to investigate in more depth yet.

- The presented unnpFOM metric showed that the main problem of all our STD systems is in calibration across different utterances. We will investigate various known normalization techniques which could improve the npFOM performance towards the unnpFOM.
- Techniques for merging overlapping detections of the WFST system, showed suboptimal performance. We will analyze the applied techniques more deeply to increase the performance of the WFST system in language-independent setups, to match the performance of the DTW system similarly as in language-dependent setups.
- Our WFST system is able to produce various lattice-based features for each detection. According to our preliminary experiments, training a statistical model with these features, has a potential to improve scores of detections and the whole system's performance as well.
- Comparing performance of an artificially selected single best example per query and five merged examples per query, we can see a considerable space for improvement. Thus we will also analyze and investigate methods for combining examples.
- For the GMM/HMM system, bottleneck features performed considerably better than 3-state phone posterior features. They are, however, not directly suitable for DTW and WFST systems. We will explore possibilities to use bottleneck features for all our QbE systems.
- Our preliminary experiments with combination of QbE systems showed a potential for performance improvement, but it has to be analyzed more deeply with real fusion techniques.
- All systems were evaluated with the non-pooled FOM metric. We did not yet pay attention to calibration of scores across queries, nor to providing hard decisions for detections of our QbE systems. These topics will have to be also investigated, as they are needed for many real world applications.

Bibliography

Abad, A., Astudillo, R. F., and Trancoso, I. 2012. The l2f spoken web search system for mediaeval 2012. In *MediaEval*.

Anguera, X. 2013. Information retrieval-based dynamic time warping. In *INTERSPEECH*. 1–5.

Barnard, E., Davel, M., van Heerden, C., Kleynhans, N., and Bali, K. 2011. Phone recognition for spoken web search. In *Proceedings of MediaEval'11*. 5–6.

Can, D. and Saraclar, M. 2011. Lattice indexing for spoken term detection. *Audio, Speech, and Language Processing, IEEE Transactions on 19,* 8, 2338–2347.

Chan, C.-A. and Lee, L.-S. 2011. Unsupervised hidden markov modeling of spoken queries for spoken term detection without speech recognition. In *INTERSPEECH*. 2141–2144.

Grézl, F. and Fousek, P. 2008. Optimizing bottle-neck features for LVCSR. In *Proceedings of ICASSP'08*. 4729–4732.

Hakkani-Tür, D., Béchet, F., Riccardi, G., and Tur, G. 2006. Beyond ASR 1-best: Using word confusion networks in spoken language understanding. *Computer Speech & Language 20,* 4, 495–514.

Hazen, T. J., Shen, W., and White, C. M. 2009. Query-by-example spoken term detection using phonetic posteriorgram templates. In *Proceedings of ASRU'09*. 421–426.

Lin, H., Stupakov, A., and Bilmes, J. 2008. Spoken keyword spotting via multi-lattice alignment. In *Proceedings of Interspeech'08*. 2191–2194.

Lin, H., Stupakov, A., and Bilmes, J. 2009. Improving multi-lattice alignment based spoken keyword spotting. In *Proceedings of ICASSP'09*. 4877–4880.

Mangu, L., Brill, E., and Stolcke, A. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language 14,* 4, 373–400.

Mangu, L., Kingsbury, B., Soltau, H., Kuo, H.-K., and Picheny, M. 2014. Efficient spoken term detection using confusion networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 7844–7848.

Mantena, G. and Anguera, X. 2013. Speed improvements to information retrieval-based dynamic time warping using hierarchical k-means clustering. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 8515–8519.

Metze, F., Anguera, X., Barnard, E., Davel, M., and Gravier, G. 2014. Language independent search in MediaEval's Spoken Web Search task. *Computer Speech & Language 28,* 5, 1066–1082.

Muscariello, A., Gravier, G., and Bimbot, F. 2011. Zero-resource audio-only spoken term detection based on a combination of template matching techniques. In *Proceedings of Interspeech'11*. 921–924.

Parada, C., Sethy, A., and Ramabhadran, B. 2009. Query-by-example Spoken Term Detection For OOV terms. *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, 404–409.

Schmalenstroeer, J., Bartek, M., and Haeb-Umbach, R. 2011. Unsupervised learning of acoustic events using dynamic time warping and hierarchical k-means++ clustering. In *INTERSPEECH*. 305–308.

Shen, W., White, C. M., and Hazen, T. J. 2009. A comparison of query-by-example methods for spoken term detection. In *Proceedings of Interspeech'09*. 2143–2146.

Szöke, I. 2010. Hybrid word-subword spoken term detection. Ph.D. thesis, Brno University of Technology, Brno, Czech Republic.

Szöke, I., Fapso, M., and Veselỳ, K. 2012. But2012 approaches for spoken web search-mediaeval 2012. In *MediaEval*. Citeseer.

Szöke, I., Tejedor, J., Fapšo, M., and Colás, J. 2011. BUT-HCTLab approaches for spoken web search - mediaeval 2011. In *Proceedings of MediaEval'11*. 11–12.

Tejedor, J., Fapšo, M., Szöke, I., Černocký, J. H., and Grézl, F. 2012. Comparison of methods for language-dependent and language-independent query-by-example spoken term detection. *ACM Trans. Inf. Syst. 30,* 3, 18:1–18:34.

Tejedor, J., Szöke, I., and Fapšo, M. 2010. Novel methods for query selection and query combination in query-by-example spoken term detection. In *Proceedings of the Searching Spontaneous Conversational Speech (SSCS'10)*. 15–20.

Velivelli, A., Zhai, C., and Huang, T. S. 2003. Audio segment retrieval using a synthesized hmm. In *Proceedings of the ACM SIGIR workshop on multimedia information retrieval, Toronto, Canada*.

Young, S. J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. 2006. *The HTK Book Version 3.4*. Cambridge University Press.

Zhang, Y. and Glass, J. R. 2011. A piecewise aggregate approximation lower-bound estimate for posteriorgram-based dynamic time warping. In *INTERSPEECH*. 1909–1912.