Vysoké učení technické v Brně
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií

Ing. Igor Szöke

Hybridní slovní a pod-slovní detekce klíčových frází

Hybrid word-subword spoken term detection

Obor: Informační technologie

Zkrácená verze disertační práce

Školitel: Doc. Dr. Ing. Jan Černocký

Klíčová slova: detekce klíčových slov, detekce frází v řeči, míry konfidence, rozpoznávání spojité řeči s velkým slovníkem, kombinovaný slovní-podslovní systém, slova mimo slovník

Keywords: keyword spotting, spoken term detection, confidence measures, large vocabulary continuous speech recognition, combined word-subword system, out-of-vocabulary

Rukopis disertační práce je uložen na Fakultě informačních technologií Vysokého učení technického v Brně, Božetěchova 2, 61266 Brno. Plný text disertační práce je k disposici na:
`http://www.fit.vutbr.cz/~szoke/publi/dis.pdf`

Contents

Chapter 1

Introduction

The research field of this thesis is spoken term detection. The corner stone of this thesis is search of out-of-vocabulary terms which are not present in dictionary of word-based speech recognizer. Also, topics as term confidence measures, weighted finite state transducers, indexing of spoken documents and phone multigram units are touched.

Short definition of important terms is placed in the following paragraph to avoid confusion of the reader of this thesis. We define the differences between `keyword`, `term`, `query`, `keyword spotting` and `spoken term detection`.

Keyword is understood as a single word within the scope of this thesis (e.g. `"IGOR"` or `"DETECTION"`). It is used within **acoustic keyword spotting** context. In fact, the keyword can be also sequence of consecutive words `"IGOR SZÖKE"` in context of *acoustic keyword spotting*. It is why these consecutive words can be processed as one keyword `"IGORSZÖKE"`.

Term is defined as one or multiple words in sequence like `"KEYWORD"`, `"KEYWORD DETECTION"` or `"THE PRESIDENT GEORGE BUSH"`. It is used within **spoken term detection** context. If the term consists of one word, there is no difference between *term* and *keyword*. For terms containing multiple words, the exact logic of how the words can be connected needs to be defined by the spoken term detector. For example, the `"KEYWORD DETECTION"` term can mean words `"KEYWORD"` and `"DETECTION"` in sequence where silence between them is shorter than 1s. Another words can be allowed between these two words. These conditions are defined in the spoken term detection system.

Query is defined as one or multiple words consisting of terms and operators `"('IGOR SZÖKE' near THESIS) and 'KEYWORD SPOTTING' not BIOLOGY"`. The operators should define the semantic information. The query is usually used in context of spoken document retrieval or information retrieval.

Keyword spotting system is a system for spotting (searching) given keywords in speech data. It understands the keyword as one object despite the number of words the keyword list might consist of. Keyword spotting system can be based on speech recognizer but it can be also "standalone" system which spots only given keywords and does not "understand" surrounding speech.

Spoken term detection system is also a system for spotting (searching) given terms in given speech data. On contrary to the *keyword spotter*, spoken term detector somehow parses and splits multiple word terms and searches for term candidates according to defined criteria (distance for example). The spoken term detection system is usually built-up on speech recognizer (and depends on it).

The topic of this thesis is aimed to **Spoken Term Detection** – STD. The STD system takes a set of terms and output of a speech recognizer and produces a list of putative hits of given term. The term is understood as sequence of one or more consecutive words. Only short silence is allowed between these particular words. Term definition is discussed more thoroughly in section 2. Our spoken term detector is based on a **large vocabulary continuous speech recognizer** – LVCSR. It takes the output of speech recognizer and provides search of terms. The speech recognizer is mainly taken "as is" and is described in chapter 3. The output list of putative hits of given term can be viewed by human or processed by a system (information retrieval or spoken document retrieval) allowing for search for more complex queries.

The complexity of spoken term detector depends on the output of speech recognizer. Such output can be a 1-*best* output (simple text string, spoken term detection is then simple text search), an N-*best* output or a graph of parallel hypothesis so called **lattice** (for definition see section 1.2). The recognizer can recognize *word units* or *subword units* (syllables, phones, etc.).

Out-of-vocabulary (OOV) words handling is also important in case of word-recognition. Words which are not present in word recognizer dictionary should be detected. Normalization is useful for scaling and shifting of term confidences. Each term should have the **confidence** normalized, so that one global threshold can be used for decision of acceptance/rejection of terms. Speed and computational requirements are also important from practical point of view.

Search accuracy depends on recognition accuracy of used speech recognizer. We need only 1-best (single string) output in a case of 100% reliable speech recognizer. Nowadays, the state of the art word recognizers achieve about 10% – 20% word error rate (WER) and about 5% – 10% *lattice word error rate* on broadcast news and *conversation telephone speech* (CTS) [12]. This gives very good search results in combination with lattice search [7]. But the language is an evolving thing and

each day many new words appear. There can be hardly a speech recognizer having all words in the dictionary. Information theory also states that the least frequent words carry most of the information. That is why we aim at out-of-vocabulary words.

The problem of OOVs can be solved by recognizing subword units (syllables or phones). The drawback of this approach is absence of strong word n-gram language model and strong acoustic model of words which are both included in large vocabulary continuous speech recognizer (*LVCSR*). That is why subword recognition does not achieve so good accuracies. Phone recognition is quite sensitive to pronunciation errors for example. These possible errors should be taken into account in the search. On the other hand, LVCSR contains only a close set of words to be recognized and word language model prefers likely word sequences off the "exotic" ones (probably carrying higher information). Also it is shown that if an OOV appears, it usually causes no 1 word error, but approximately $2-4$ word errors [2]. This is a justification of an investigation into subword recognition.

1.1 Spoken term detection

The generic scheme of a spoken term detection system is in figure 1.1. The spoken term detection system is built on speech recognizer, which usually encapsulates also the feature extraction. The speech recognizer produces textual strings or so-called lattices (figure 1.2) which contain transcribed speech in words labels. The lattices are searched for the given terms or keywords.

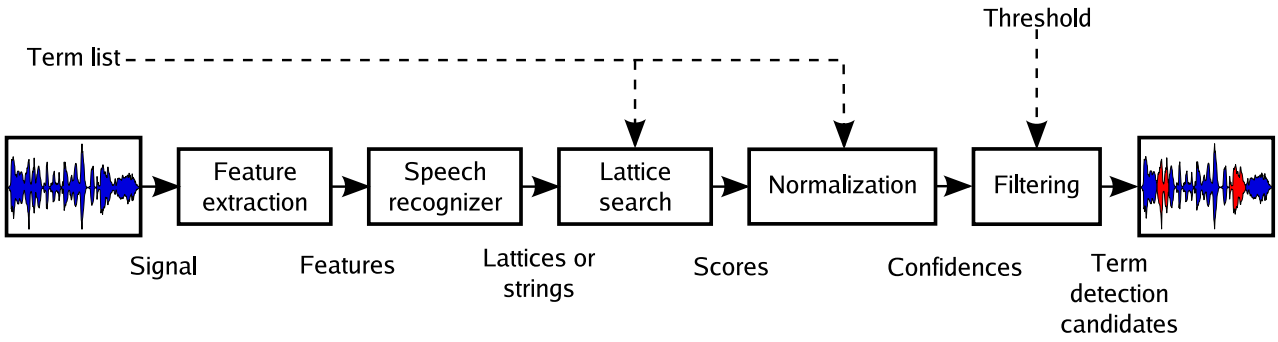

Figure 1.1: General scheme of spoken term detector.

Spoken term detection (indirect keyword spotting) is based on the output of a speech recognizer. It is a two step method where the first step consists of the time consuming speech recognition and the second one consists of a fast spoken term/keyword search. The method inherits main characteristics of the recognizer

used. Input term/keyword must be converted to a sequence of units similar to recognizer's output units (e.g. words, syllables, phones, etc.). Then the sequence is searched in the output of the recognizer. The recognizer (usually the slowest step of whole STD) is run only once. The STD or keyword spotter is run each time a term or keyword has to be found. In comparison to the acoustic keyword spotting, the search is very fast because it is done over "textual data" (output of speech recognizer). Advantages of STD are the speed of search and detection accuracy (depends on recognizer's accuracy). Searching speed can be optimized by techniques known in information retrieval, such as inverted indices, caching etc. to achieve searching times less than $10^{-3}s/hr/term$. The disadvantages are off-line processing (especially LVCSR is complex and time consuming) and closed unit vocabulary. The recognizer has finite and closed vocabulary of units it can recognize. Once the recognition is done, the spoken term detector will "find" only units which were recognized by the recognizer. This is a drawback if a word recognizer is used. STD approach can be split according to used recognizer to word-based and subword-based. The word-based STD has very high accuracy (having phone models "organized" in words and strong word language model) but limited vocabulary. The subword-based STD approach has unlimited vocabulary (search word must be converted to a sequence of subword units) but has lower accuracy (missing word acoustic models and word language model).

In STD, we "ask" for the posterior probability $p(term_{t_b}^{t_e})$ of occurrence of the term $term$ from time t_b to time t_e. A sequence of units $\mathbf{w}$ is constrained to $\mathbf{w}(term_{t_b}^{t_e})$ which contain the term in given time:

$$\hat{\mathbf{w}}(term_{t_b}^{t_e}) = \underset{\mathbf{w}(term_{t_b}^{t_e}) \in \mathcal{W}(term_{t_b}^{t_e})}{\arg\max} p(\mathbf{w}(term_{t_b}^{t_e})|\mathcal{D}), \tag{1.1}$$

where $\mathcal{W}(term_{t_b}^{t_e})$ is the set of all permissible sentences having the term in defined time and $\mathcal{D}$ is the observed data. Applying the Bayes formula, we get

$$\hat{\mathbf{w}}(term_{t_b}^{t_e}) = \underset{\mathbf{w}(term_{t_b}^{t_e}) \in \mathcal{W}(term_{t_b}^{t_e})}{\arg\max} \frac{p(\mathcal{D}|\mathbf{w}(term_{t_b}^{t_e}))p(\mathbf{w}(term_{t_b}^{t_e}))}{\sum_{\mathbf{w}' \in \mathcal{W}} p(\mathcal{D}|\mathbf{w}')p(\mathbf{w}')}. \tag{1.2}$$

In practice, direct implementation of formula 1.2 is difficult. We do not know the time of occurrence t_b and t_e of the term $term$. Again, an approximation must be used to hypothesize t_b and t_e. The time of the term can be suggested from $\mathcal{W}$. To avoid having size of $\mathcal{W}$ infinite, $\mathcal{W}$ is approximated by lattice.

So, the real spoken detection task has two steps. The set of the most likely hypothesis $\mathcal{W}'$ is generated. Then occurrences of searched terms are found in $\mathcal{W}'$

and estimation of term posterior probability $p(term_{t_b}^{t_e})$ is:

$$p(term_{t_b}^{t_e}) = \frac{p(\mathcal{D}|\mathbf{w}(term_{t_b}^{t_e}))p(\mathbf{w}(term_{t_b}^{t_e}))}{\sum_{\mathbf{w}' \in \mathcal{W}'} p(\mathcal{D}|\mathbf{w}')p(\mathbf{w}')}. \tag{1.3}$$

1.2 Search in lattice

This section presents the "implementation" of the calculation of term posterior probability stated in equation 1.3 in the previous section. Lattice (figure 1.2) are nowadays used as the multiple hypothesis output of speech recognizer.

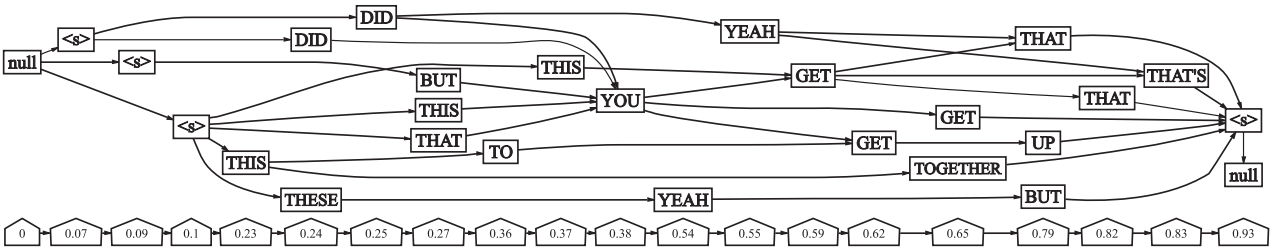

Figure 1.2: An example of word lattice. X-axis represents time.

The lattice is an acyclic oriented graph. Each node n represents a time. An arc a connects two nodes n_1, n_2 and represents a speech unit[1] $u = U(a)$ and set of two likelihoods $L(a)$ (acoustic $L_{Ac}(a)$ and language $L_{LM}(a)$). Start time $t_b(a)$ and end time $t_e(a)$ of arc a representing unit $U(a)$ correspond to the time of start node $t(n_b(a))$ and end node $t(n_e(a))$ of the arc a:

$$\begin{aligned} t(n_b(a)) &= t_b(a) \\ t(n_e(a)) &= t_e(a). \end{aligned}$$

The $L_{Ac}(a) \propto p(\mathcal{D}|\mathbf{w}(a_{t_e}^{t_b}))$ and $L_{LM}(a) \propto p(\mathbf{w}(a_{t_e}^{t_b}))$.

The best hypothesis (the most likely path) can be derived from lattice. The best path through the lattice is also known as 1-*best* or *string* output. N most likely paths through the lattice are known as N-*best* output. Lattice can be understood as compact representation of the N-best output where the N is a large number.

Searching for a term in the string output (1-best) is straightforward. An algorithm goes through the string of units and compares each term to a sequence of units. If the comparison is successful, time boundaries and likelihood of units are stored to a list of term detections. The drawback of this approach is the absence of normalization.

[1] Another possibility is to represent speech unit as end node of the arc, then the arc represents only time information and likelihoods.

Term scores, which are derived from likelihoods of units ($L(u)$), are sensitive to background noises. The term detector is not robust in this case.

On the other hand, searching for the term in the lattice is more robust. Having the lattice, we have the $\mathcal{W}'$ and we can estimate the posterior probability of term according to equation 1.3. The posterior probability gives confidence of term for particular occurrence of term (represented by arc a) in time $t_b(a), t_e(a)$.

However, one more problem should be solved. Assume, that we also hypothesized occurrence a' of the term, which is slightly shifted but still overlapped with the original one. The problem is: is the probability of the original occurrence affected by the fact that several overlapped occurrences of the same term exist? This leads to "alternative" formula estimating the posterior probability of the term in time t: $t(term) = term_t$. The occurrence of term in time $t(term)$ is defined by condition $t_b(term) \leq t(term) \leq t_e(term)$.

These two points of view are defined in this thesis in the following way:

1. The **term score**. The term score is the posterior probability $p(term_{t_b}^{t_e})$ of particular term hypothesis in the lattice from time t_b to time t_e (figure 1.3). It does not consider other overlapped occurrences of the term in the time.

2. The **term confidence**. On the other hand, term confidence is the posterior probability $c(term_t) = p(term_t)$ of existence of the term in the lattice at given time t (figure 1.3). It takes into account several overlapped particular term hypothesis in the lattice.

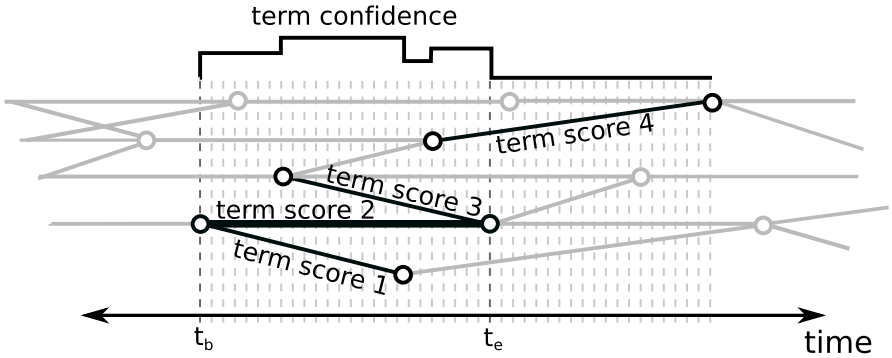

Figure 1.3: Example of a term occurrences in a lattice. "Term scores" denote different values of the posterior probability $p(term_{t_b}^{t_e})$ for particular term occurrence. "Term confidence" denotes evolution of posterior probability $p(term_t)$ of existence of the term in the lattice at given time t.

Chapter 2

Evaluation

Well defined evaluation data is important for objective evaluation and comparison of different systems. Unfortunately, each published spoken term detection system was evaluated on different data and with different term set. There were 4 TREC NIST evaluations in years $1997 - 2000$. The TREC evaluations had only partial overlap with spoken term detection task. The goal of TREC was *Spoken Document Retrieval* (TREC-SDR) on broadcast news. The broadcast news recordings were recognized by *Automatic Speech Recognizer* and then processed by *Document Retrieval* system. The goal was to find the relevant document, but not to find all term occurrences. The TREC-SDR was declared a solved problem at TREC-9 in 2000. There were no spoken term detection evaluations organized by NIST from year 2000 till 2006. New evaluation track was announced by NIST in 2006. It was called *Spoken Term Detection* (STD) evaluation[1].

The goal of the first NIST STD (2006) evaluation was to explore promising new ideas in spoken term detection and measuring the performance of this technology [6]. The spoken term detection system should consist of two parts:

- The first part is an *indexing sub-system*. It processes all input speech data (audio signal) into indices. This step can take longer time (hours of processing time per hour of speech data) and is run on the data only once.
- The second part is a *search sub-system*. It should find a given term as fast as possible (milliseconds of processing time per one term) in the indices.

The STD 2006 evaluation task was to find all of the occurrences of a specified term in a given corpus of speech data. The "term" is a sequence of one, two, three or four words. The words in a term have to be said by the same speaker, channel and file. The gap between adjacent words must not be longer than 0.5s. Terms are

[1] `http://www.itl.nist.gov/iad/mig/tests/std/2006/index.html`

specified only by orthographic representation so ”wind” (moving air) will match ”wind” (twist) but ”cat” will not match ”catalog”. The evaluations ran for 3 different domains and 3 languages, see table 2.1.

Domain \ Language	English	Arabic	Mandarin
Broadcast News (BCN)	$\sim$ 3 hours	$\sim$ 1 hour	$\sim$ 1 hour
Telephone Conversations (CTS)	$\sim$ 3 hours	$\sim$ 1 hour	$\sim$ 1 hour
Round-table Meetings (MTG)	$\sim$ 2 hours	No	No

Table 2.1: Durations of indexed audio for both, the DevSet and the EvalSet.

NIST provides three data sets. A *Development set* (**DevSet**), a *Dry Run set* (**DryRunSet**) and an *Evaluation set* (**EvalSet**). The *DevSet* was offered for system development. It contains speech data, reference transcripts and a list of 1099 terms. The *DryRunSet* differs from DevSet only in different term list (1099 terms). The dry run was just for evaluation of participant competence to use NIST scoring tools and to generate correct result files.

The *EvalSet* contains different speech data and a different term list (1099 terms). Unfortunately, NIST decided not to publish reference transcriptions. The EvalSet will be reused for next evaluations due to lack of speech data. This complicates evaluation of STD systems, because there is only the DevSet.

Using *round-table meeting data* (MTG) and *conversational telephone speech* (CTS) brings more objectivity, because it is more natural form of speech (in comparison to *broadcast news data* (BCN)). Meeting or telephone dialog participants speak informally and the speech is spontaneous containing lots of hesitations, crosstalk, smacks and background noises. This data is closer to the security domain.

The CTS data of NIST STD 2006 DevSet is used in this thesis for STD evaluation. As because the speech recognizer (chapter 3) is taken as a “black box” and NIST released only the *DevSet*, several system coefficients are tuned on the DevSet: unit (word, phone or multigram) insertion penalty and language model or acoustic model scaling factors. We assume that tuning of these parameters has no impact on the correctness of the results and conclusions.

2.1 Term set modification and vocabulary reduction

The original term set for English part of 2006 NIST STD evaluations is not representative for our experiments, because it contains low number of out-of-vocabulary (OOV) words. We decided to make several changes to the STD term list and our speech recognizer vocabulary to achieve higher OOV rate. First of all, all terms

containing true OOV words or 1 phone long[2] were omitted. The 1 phone long term is not a big problem for word-based STD, but serious problem for phone based STD (huge number of detections).

Then a set of "artificial" OOV words is defined – these are originally in the recognition vocabulary, but deleted for future experiments to create more OOVs. Their selection is done in the following way: Word counts are collected over the *DevSet*. Based on these counts, a suitable set of OOVs was selected: The word had to have several occurrences, but generally less than 10. Only 5 OOVs have more than 10 occurrences. In total, 880 words were deleted in this way, of which 440 do appear in NIST dev-set transcriptions. Another 440 words which do not appear in the transcriptions were simply selected from the LVCSR vocabulary. They are of no use in this these, but reserved for future work.

A limited LVCSR system was created (denoted by **WRDRED** which means "**reduced vocabulary**") where these 880 words were omitted from the vocabulary. This system has reasonably high OOV rate on the NIST STD06 DevSet. The term set has 975 terms of which 481 are in-vocabulary (IV) terms and 494 are out-of-vocabulary OOV terms (terms containing at least one OOV) for the reduced system. The number of occurrences is 4737 and 196 for IV and OOV terms respectively. We can detect all the "artificial" OOV terms by the original **full vocabulary** LVCSR (denoted as **WRD**) and evaluate the "oracle" OOV term detection accuracy.

Reference transcription of the NIST STD 2006 DevSet has 32002 tokens. Defined "artificial" OOVs appear 799 times in the corpus. So the OOV rate is 2.5%, which is close to real tasks.

Table 2.1 summarizes the numbers of terms and term occurrences for different term length and data types in DevSet.

2.2 Spoken Term Detection evaluation metrics

This section presents evaluation metrics which are used for spoken term detection and keyword spotting task. Each detected term has a confidence attached. The confidence is a continuous value quantifying, how sure the spoken term detector is about the detection of the term. Some users of spoken term detection application expect hard *YES/NO* decision whether a term is present or not. Another users expect only *YES* decision (rising of an alarm). *NO* decision is the complement to *YES* decision over input speech data. Confidence thresholding is used mapping of confidence to hard binary YES/NO decision. Let us assume that the term confidence is based

[2] term "A."

Term			CTS			
length [words]	count		count		terms occur.	
	IV	OOV	IV	OOV	IV	OOV
1	309	245	214	76	4640	156
2	149	197	42	30	92	34
3	21	45	5	5	5	5
4	2	7	0	1	0	1
sum	481	494	261	112	4737	196

Table 2.2: Distribution of terms for `reduced LVCSR 50k vocabulary` – **WRDRED** system. The second and third columns give the numbers of IV or OOV terms in the term list. The next two columns summarize the numbers of the terms appearing in the CTS. The last two columns represent the numbers of occurrences of IV and OOV terms in CTS set. The true OOV terms and 1 phone long terms are omitted.

on term posterior probability. The higher confidence value the higher probability of correct term detection. Let us set the threshold thr to a certain value. The term confidence $c(term_t)$ thresholding is defined by:

$$Decision(c(term_t), thr) = \begin{cases} YES, & c(term_t) > thr \\ NO, & c(term_t) <= thr \end{cases} \tag{2.1}$$

where $Decision$ function returns the hard decision whether the term is found or not. Several cases can occur in comparison of detected terms against reference detections (transcription):

1. The decision is *YES* (alarm is raised) and there is a reference term *overlapped* with the detected term in time. This case is denoted as **HIT**. We want to maximize the number of hits.

2. The decision is *YES* (alarm is raised) and there is no reference term *overlapped* with the detected term in time. This case is denoted as **false alarm** – **FA**. We want to minimize the number of false alarms.

3. There is a reference term in utterance but no *overlapped* term is detected at that place or *overlapped* detected term is marked by *NO* decision (no alarm is raised at the same time). This case is denoted as a **false rejection** – **FR** or a **MISS**. We need also to minimize the number of false rejections.

The definition of "*overlapped*" for reference and detected term varies for different evaluation metrics. In our case, the mid-point of detected term is less than or equal to 0.5s from the time span of reference term for **term weighted value – TWV** metric [6] used in NIST STD 2006 evaluations. If more detections overlap

with one reference, only one is considered as HIT and the others are considered as FAs (figure 2.1).

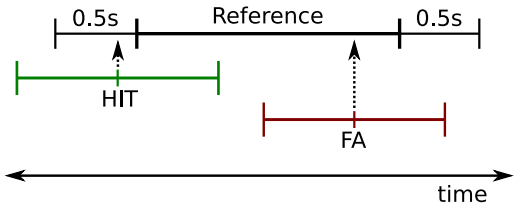

Figure 2.1: Example of HIT and reference overlap defined by NIST for STD evaluation and TWV metric. If two detections overlap one reference, only one is considered as HIT and the other is considered as FA.

The level of threshold can be set for each term. More HITs (and also more FAs) and less FRs are received by lowering the threshold, less HITs (and also less FAs) and more FRs are received by increasing the threshold. The numbers of HITs and FAs are correlated and as number of HITs rises so does the number of FAs. The user must set the threshold to obtain the desired system behavior (high number of HITs or low number of FAs). The accuracy of a term detection system rises as rises the separability of HITs and FAs. A system will have 100% of HITs and 0% of FAs for a certain threshold in an ideal case of the best accuracy. Setting of optimal threshold is nontrivial especially if one global threshold applied over a large set of terms.

The probability of correct detections p_{HIT}, false rejections p_{MISS} and incorrect detections p_{FA} can be calculated by the following formulas. Let us denote:

- $term$ searched term
- thr set threshold
- $N_{target}(term)$ the number of all correct occurrences of $term$ in the data set
- $N_{HIT}(term, thr)$ the number of detections having $Decision(c(term), thr) = YES$ which are classified as HIT
- $N_{nontarget}(term)$ the number of all non-occurrences of $term$ in the data set
- $N_{FA}(term, thr)$ the number of detections having $Decision(c(term), thr) = YES$ which are classified as FA

The $N_{nontarget}(term)$ means all places, where false alarms of the $term$ can occur.

The probability of HIT is defined as:

$$p_{HIT}(term, thr) = \frac{N_{HIT}(term, thr)}{N_{target}(term)} \tag{2.2}$$

The probability of MISS is defined as:

$$p_{MISS}(term, thr) = 1 - p_{HIT}(term, thr) = 1 - \frac{N_{HIT}(term, thr)}{N_{target}(term)} \tag{2.3}$$

The probability of False Alarm is defined as:

$$p_{FA}(term, thr) = \frac{N_{FA}(term, thr)}{N_{nontarget}(term)} \tag{2.4}$$

The performance of spoken term detection system is defined by the trade-off between p_{HIT} and p_{FA}. As this is not a scalar value, different systems can not be easily compared according to p_{HIT} and p_{FA}. That is why several metrics have been proposed for calculation of one scalar value from p_{HIT} and p_{FA}. Some of them are used for comparison of detectors in this thesis. Their brief description and definition follows in section below.

2.2.1 Upper bound term-weighted value – UBTWV

One feature of TWV metric is its one global threshold for all terms. This is good for evaluation for end-user environment. On the other hand, it leads to uncertainty in comparison of different experimental system setups. We do not know if the difference is caused by different systems or different normalization and global threshold estimation. This is reason for our definition of **Upper Bound TWV** (UBTWV). The difference to TWV is in individual threshold per each term. The ideal threshold for each term is found to maximize term's TWV:

$$thr_{ideal}(term) = \arg\max_{thr} TWV(term, thr), \tag{2.5}$$

and UBTWV is then defined as:

$$UBTWV = 1 - \underset{term}{average}\{p_{MISS}(term, thr_{ideal}(term)) + \beta p_{FA}(term, thr_{ideal}(term))\} \tag{2.6}$$

This is equivalent to shifting the score of each term, so that maximum $TWV(term)$ is obtained at threshold 0.0. Two systems can be compared by UBTWV without any influence of normalization and ideal threshold level estimation in systems producing TWV score. The *actual* and *maximal* values are equal for UBTWV and both are denoted by **UBTWV**. However, due to the fact that each term has its ideal threshold, DET curve for such ideal system has not much sense. Only the point corresponding to the ideal threshold is important. This point is supplied by the UBTWV. That is why only UBTWV values without DET curves are reported in this thesis.

Chapter 3

Word recognition

This section deals with the description of our large vocabulary continuous speech recognition system (LVCSR) used for experiments stated in this thesis. Presented LVCSR is a state-of-the-art system derived from AMI LVCSR[1] [10]. The AMI LVCSR system was slightly modified and used in the NIST STD 2006 evaluation. The decoder was changed from HTK *HDecode* to "in-house" STK *SVite* in the third (final) pass and produced lattices are directly used for STD. In the AMI LVCSR, the lattices were expanded by fourgram language model and confusion networks were applied.

3.1 The recognizer

The input data (conversational telephone speech) is first converted to linear coding 16-bits per sample and 8 kHz. The data is then segmented to speech/silence according to energy in channels and by a neural net based phone recognizer [14]. All phone classes are linked to "speech" class.

The data is split into shorter segments on silences (output of speech/non-speech detector) longer than 0.5s. If the speaker changes, the data is also split. Segments longer than 1 minute are split into 2 parts in silence closest to the center of the segment. This is done to overcome long segments and accompanying problems during decoding (long decoding time and high memory consumption).

The large vocabulary continuous speech recognition system (LVCSR) system used in this thesis is a simplified version of AMI LVCSR system used for NIST RT 2006 evaluations [9]. The system operates in 3 passes (figure 3.1):

In the **first pass** – **P1**, the front-end converts the segmented recordings into feature streams, with vectors comprised of 12 *Mel-Frequency Perceptual Linear Prediction* (MF-PLP) features and raw log energy. First and second order derivatives

[1]The LVCSR was developed in cooperation with AMI-project partners, see `http://www.amiproject.org`.

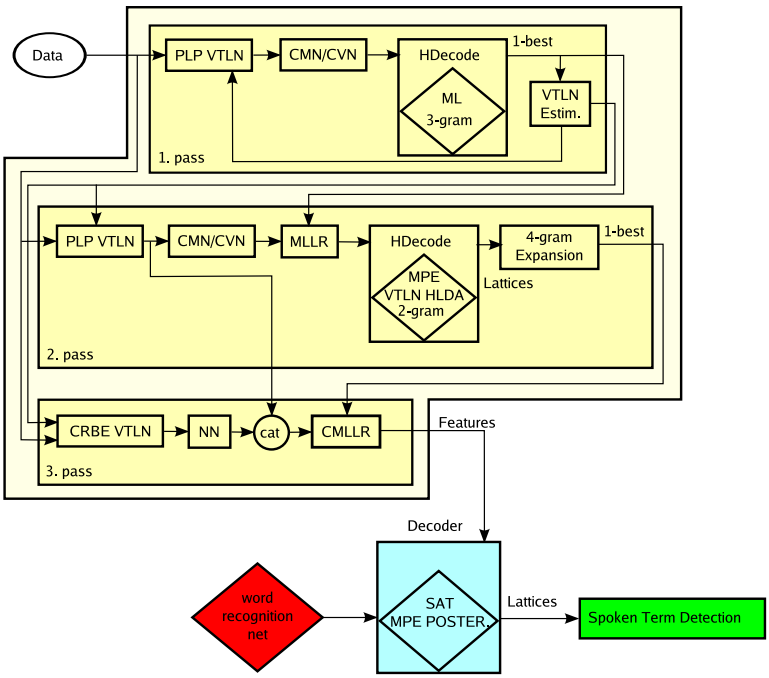

Figure 3.1: Schema of 3-pass recognition system used in this thesis. The system is derived from AMI LVCSR.

are added. After, *Cepstral Mean and Variance Normalization* (CMN/CVN) is performed on a per channel basis. The first decoding pass yields initial transcripts that are subsequently used for estimation of *Vocal Tract Length Normalization* (VTLN) warp factors. The feature vectors and CMN and CVN are recomputed after the application of VTLN.

The **second pass** – **P2** processes the new features and its output is used to adapt models with *Maximum Likelihood Linear Regression* (MLLR). Bigram lattices are produced and re-scored by trigram and fourgram language model.

In the **third pass** – **P3**, posterior features [8] are generated. The output from the second pass is used to adapt models with *Constrained MLLR* (CMLLR) and MLLR. *In the original AMI LVCSR*, bigram lattices were produced by HDecode decoder and re-scored by fourgram language model. *In this thesis*, the output of the third pass are the features which are processed by SVite decoder.

3.2 Baseline word recognition systems

Selected LVCSR system parameters are *LM pruning* 2×10^{-9}, $Pr = 260$ (beam pruning) and $MAM = 5000$ (maximum active models). System with these parameters achieved very good accuracy, small size of lattices and low decoding time. It is important to note, that this was original AMI CTS system with closed vocabulary

language model. This baseline closed vocabulary LVCSR system was denoted as *WRD*. Word recognition system with reduced vocabulary (derived from *WRD* system) was used in following chapter 4. It was *WRD* system where 880 words were omitted from the vocabulary. Details of vocabulary reduction were given in section 2.1. This system was denoted as *WRDRED*. Both these baseline systems were compared in the upper part of table 3.1.

However, open vocabulary language model is needed for our later experiments in chapter 5. The open vocabulary language model had to be trained "from scratch" (not only by omitting 880 words) in order to correctly estimate the probabilities of the "out-of-vocabulary" symbol <unk>.

To make the systems comparable, we created *WRDforHYB* system, which should be comparable to baseline open vocabulary word recognition systems presented in chapter 5. The *WRDforHYB* system used $Pr = 220$ and LM pruning 1×10^{-8} because it is close to the open vocabulary LM in terms of number of bigrams: The accuracies of *WRDforHYB* system are presented in the bottom part of table 3.1.

System	Decoder Pruning	LM Pruning	wrdSIZE	WAC	WLAC	Word UBTWV		
						ALL	IV	OOV
WRD	260	2×10^{-9}	0.510	70.78	88.22	0.795	0.777	0.838
WRDRED	260	2×10^{-9}	0.507	68.41	85.75	0.522	0.747	0.000
WRDforHYB	220	1×10^{-8}	0.252	69.04	83.21	0.738	0.734	0.746

Table 3.1: Comparison of lattice size, word accuracy, word lattice accuracy and UBTWV of different baseline LVCSR systems. WRD is the "full" vocabulary baseline, WRDRED is the reduced vocabulary baseline and WRDforHYB is "full" vocabulary baseline comparable to open vocabulary LVCSR in terms of LM size and decoder pruning.

Chapter 4

Subword recognition – phone multigrams

This chapter deals with theoretical description and experimental evaluation of multigram units.

Examples of other subword units [13] besides phones are *syllables*, *phone n-grams*, *phone multigrams*, *broad phone classes*. All these units are based on phones. Phone recognition and search using such units has its advantages and drawbacks. The advantage of phone recognition (using simple phone loop) is its relative simplicity and presence of minimum constraints. Produced phone string precisely reflects a spoken word or term. This holds if the acoustic models are highly accurate and the word (or term) was uttered correctly. Then phone string produced by the phone recognizer perfectly matches the searched phonetic word form. But these two conditions are rarely fulfilled.

The drawbacks of phones are the following: If the model is not 100% accurate, the speaker does not pronounce well, or there is a background noise, recognized phones do not match the speech well. Also, decoding from free phone loop with higher order of n-gram language model is computationally more expensive than the decoding from LVCSR network[1]. Longer units should be more robust for incorrect pronunciation of a term too. Finally, phone n-grams with fixed length n must be used for indexing of phone strings or lattices. The optimal length of phone n-grams was found to be 3 in [13]. In our prior work [15], we have also used sequences of overlapped 3-grams for search. However, out-of-vocabulary words shorter than 3 phones were dropped.

The disadvantage of the fixed length sequences is that the frequencies of phone sequences are not taken into account. Some phone trigrams are more frequent than the others. Variable length sequences can be used to overcome this problem: a rare sequence is split into more frequent shorter sequences while a frequent sequence

[1] All phone acoustic models must be evaluated in the phone loop approach, while words and word language model reduce the search space significantly. This leads to evaluation of limited set of phone acoustic models and lower computational requirement.

can be represented as the whole unit. The other advantage is that variable length phone units can reflect word sequences and compensate for missing word language model.

4.1 Definition of multigrams

Variable length sequences of phones are denoted as phone multigrams. The multigram language model was proposed by Deligne et al. [5]. Multigram model is a statistical model having sequences with variable number of units. The definition of multigram model and its parameter estimation follows:

Let $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ denote a string of N units, and let $\mathbf{s}$ denote a possible segmentation of $\mathbf{w}$ into q sequences $q \leq N$ of units $\mathbf{s} = \{s_1, s_2, \ldots, s_q\}$. The n-multigram model computes the joint likelihood $L(\mathbf{w}, \mathbf{s})$ of the corpus $\mathbf{w}$ associated to segmentation $\mathbf{s}$ as the product of the probabilities p of the successive sequences, each of them having a maximum length of n:

$$L(\mathbf{w}, \mathbf{s}) = \prod_{i=1}^{q} p(s_i) \tag{4.1}$$

Denoting as $\mathcal{S}$ the set of all possible segmentations of $\mathbf{w}$ into sequences of units, the likelihood of $\mathbf{w}$ is:

$$L_{mgr}^{best}(\mathbf{w}) = \max_{\mathbf{s} \in \mathcal{S}} L(\mathbf{w}, \mathbf{s}) \tag{4.2}$$

A n-multigram model is fully defined by a set of parameters $\mathcal{P}$ consisting of the probability of each unit sequence $s_i \in \mathcal{D}$ in a dictionary $\mathcal{D} = \{s_1, s_2, \ldots s_m\}$ that contains all the sequences which can be formed by combination of $1, 2, \ldots, n$ units:

$$\mathcal{P} = (p_i)_{i=1}^{m} \quad \text{where} \quad p_i = p(s_i) \quad \text{and} \quad \sum_{i=1}^{m} p_i = 1 \tag{4.3}$$

Maximum likelihood estimates of $\mathcal{P}$ can be computed through Viterbi algorithm iteratively. Let $\mathbf{s}^{*(k)}$ denote the most likely segmentation of $\mathbf{w}$ with given parameters $\mathcal{P}^k$ at iteration k:

$$\mathbf{s}^{*(k)} = \arg\max_{\mathbf{s} \in \mathcal{S}} L(\mathbf{s}|\mathbf{w}, \mathcal{P}^k) \tag{4.4}$$

According to [5], the re-estimation formula of i^{th} parameter (sequence) at iteration

$k+1$ is intuitive:

$$\mathcal{P}_i^{k+1} = \frac{c\big(s_i, \mathbf{s}^{*(k)}\big)}{c\big(\mathbf{s}^{*(k)}\big)}, \tag{4.5}$$

where $c(s_i, \mathbf{s})$ is the number of occurrences of sequence s_i in segmentation $\mathbf{s}$ and $c(\mathbf{s})$ is the total number of sequences in $\mathbf{s}$.

The set of parameters $\mathcal{P}$ is initialized with the relative frequencies of all occurrences of units up to length n in the training corpus. To avoid overlearning, it is advantageous to discard low probable sequences: by setting $p_i = 0$ to all $c(s_i) \leq c_0$. The c_0 parameter is denoted as **multigram pruning parameter**. Sequences of length $n = 1$ are excluded from pruning to ensure that each sequence is segmentable. If a unit with length $n = 1$ has 0 occurrences in $\mathbf{s}$, then it's probability is set to a very low number.

When the set of parameters $\mathcal{P}$ is estimated, any phone string can be segmented into sequence of phone multigrams. The process of segmentation is straightforward. All possible segmentations, according to the inventory of phone multigrams, are created. Then, probability of each segmentation is evaluated according to the probabilities of multigram units. The best (most probable) segmentation is considered as the segmentation of given phone string by the set of phone multigrams. The process of phone string segmentation to phone multigrams is implemented by the Viterbi algorithm.

4.2 Constrained multigram units

The baseline process of building multigram unit inventory is without any constraints (denoted **xwrd**). The corpus of phone strings is taken as is. An example of an utterance segmented by such unconstrained units is in table 4.1 line 2. A multigram unit can be placed across word boundaries and also across silences (`sil`). Incorporation of word boundaries (cross-word multigrams) into multigram units means, that multigrams also somehow reflect the word language model. The question is whether this is good or not. The same question can be asked about the silence `sil`. By incorporating silence into multigrams, the units are learned to remember parts of speech where silence is usual and where it is not. Two experiments with constrained training of multigram inventory are done to evaluate the influence of cross word multigrams and silence inside multigram units:

word	sil YEAH I MEAN IT IS sil INTERESTING									
xwrd	sil-y-eh-ax		ay-m-iy-n		ih-t-ih-z-sil			ih-n-t-ax-r		eh-s-t-ih-ng
nosil	sil	y-eh-ax	ay-m-iy-n		ih-t-ih-z		sil	ih-n-t-ax-r		eh-s-t-ih-ng
noxwrd	*sil*	*y-eh-ax*	*ay*	*m-iy-n*	*ih-t*	*ih-z*	*sil*	*ih-n	t-ax-r-eh-s	t-ih-ng*

Table 4.1: Examples of different multigram segmentations. The first line is word transcript. The second line is unconstrained multigram segmentation. The third line is constrained multigram segmentation where silence is forbidden inside a multigram unit. The fourth line is constrained multigram segmentation where silence and word boundary * are forbidden inside a multigram unit.

Unit	System	LM n-gram	PAC	UBTWV			SIZE
				ALL	IV	OOV	
Word	WRDRED	2	-	0.514	0.734	0.000	0.56w
Word	WRDREDtoPHN	2	65.40	0.540	0.554	0.508	4.34p
Phone	LnoOOV	3	59.66	0.483	0.453	0.552	6.38p
Mgram	xwrd	3	65.25	0.559	0.552	0.577	1.4w/3.6p
Mgram	nosil	3	**65.42**	0.584	0.578	**0.597**	1.2w/4.1p
Mgram	noxwrd	3	65.10	**0.630**	**0.647**	0.593	1.7w/3.7p

Table 4.2: Comparison of word, phone and multigram systems from phone accuracy, lattice size and Word, Mgram and Phone UBTWVpoint of view. 0.56w means wrdSIZE and 4.34p means phnSIZE.

4.2.1 No silence in multigram

Inventories of multigram units which do not contain silence are trained in this experiment (denoted **nosil**). The unigram `sil` is the only one multigram unit which contains silence. This is needed to make utterances segmentable. An example of utterance segmented by this method is in table 4.1 line 3. Building of this nosil multigram inventory is done by a modification in the first step of multigram training procedure. After the statistics of all n-grams appearing in the training corpus are collected, all n-gram units containing `sil` are omitted (except the unigram `sil`). Then, the initial probabilities of units are re-normalized and the iterative training algorithm is run.

4.2.2 Non-cross-word multigrams

In this experiment, word boundaries are marked in the training corpus, and the following rule is incorporated into the training algorithm: word boundary will appear at most at the beginning or at the end of a multigram unit. Only two units with the word boundary marker can be put besides each other during the segmentation. If the first unit contains word boundary marker at the end, then the following boundary must contain the word boundary marker at the beginning. This system is denoted as **noxwrd**. An example of utterance segmented by noxwrd multigrams is in table 4.1 line 4. The word boundary marker is denoted by a star symbol.

4.3 Conclusion

Table 4.2 compares *word*, *phone* and *phone multigram* based systems from *phone* and *spoken term detection* accuracy point of view. The WRDREDtoPHN is the WRDRED LVCSR switched to produce phone lattices. The best phone accuracy is achieved by the multigram nosil constrained system. However, better STD accuracy is achieved by the noxwrd constrained multigrams. It is important to mention that multigram lattices are significantly smaller and the recognition network is approximately of the same size compared to phone system. The multigram system has maximal multigram length $l_{mgram} = 5$ and multigram pruning $c_0 = 50$. The terms are segmented only to 1-best multigram variant.

Chapter 5

Combined word-subword spoken term detection

We investigate into the use of different combination of word and subword STD systems. Let us have a term `"Igor Szöke"`. The term is first split into in-vocabulary (IV) and out-of-vocabulary (OOV) parts. Let us assume, that the name `Igor` is in-vocabulary word `igor` and the surname `Szöke` is an out-of-vocabulary word. If we choose phones as the subword units, the out-of-vocabulary part is decomposed into sequence of phones `s eh k eh`. The combination of a word and subword based spoken term detection is needed to spot both, in-vocabulary and out-of-vocabulary parts of the term.

The *word recognizer* is considered as a strong recognizer. It has strong acoustic model (word models) and language model (word bigrams). The *subword recognizer* is considered as a weak recognizer. It has weak acoustic model (phone or phone multigram units) and relatively weak language model (phone n-grams).

The combination of word and subword recognizer should allow to traverse between words and subwords in any time. If traversing penalties and other parameters are set correctly, the word part of the recognizer should well represent in-vocabulary speech. Out-of-vocabulary parts of speech may be highly unlikely for the strong word recognizer. However these OOV parts are not so unlikely for the subword part of the recognizer. This leads the recognizer to switch from the word part to the subword part. The result is the hybrid word-subword lattice where OOV parts of speech are represented by phone sequences and IV parts of speech by word sequences.

On contrary to Bazzi [1], we aim at the *investigation of STD accuracy* and the *practical application* for searching in spoken documents. We fully use the information produced by the OOV (subword) model for search of the OOV terms. We evaluate the accuracy of STD and word accuracy. We investigate in more depth “which subword model should be the best”:

- The impact of subword model and hybrid network scaling parameters to the accuracy.
- The speed of the system and size of the index.
- Search for the system configuration suitable for practical use.
- Evaluation of the hybrid system in conjunction with indexing and search engine for spoken term detection.

5.1 Building combined word-subword hybrid recognition network

We use the same decoder (*SVite* from STK toolkit) as is used for the baseline experiments in word and phone recognition (chapters 3, 4). Because the *SVite* is a static decoder, the hybrid decoding is possible by **modification of recognition network**. No other changes are needed in the decoder.

The hybrid word-subword recognition network is built in similar way as the word recognition network. Only the language model automaton G and the lexicon L are modified in the composition:

$$H \circ C \circ L \circ G, \tag{5.1}$$

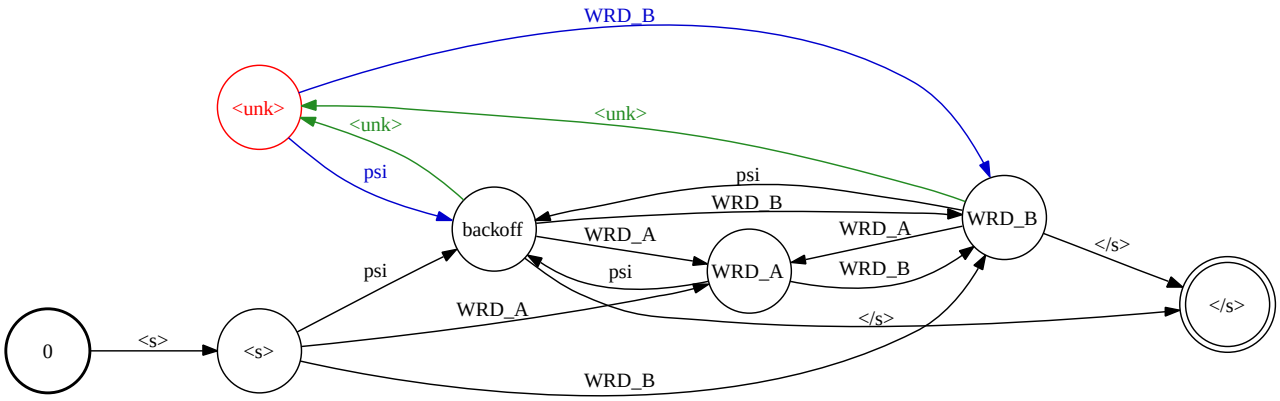

Figure 5.1: Example of open vocabulary language model. The <unk> states for the out-of-vocabulary words.

The word language model represented by WFSA G is created as open vocabulary language model and contains an <unk> symbol. The <unk> symbol represents any out-of-vocabulary word, see figure 5.1. The new open-vocabulary language model represented by WFSA in denoted as G_{word}. This <unk> symbol is substituted by a subword language model (figure 5.2). The subword language model is converted to

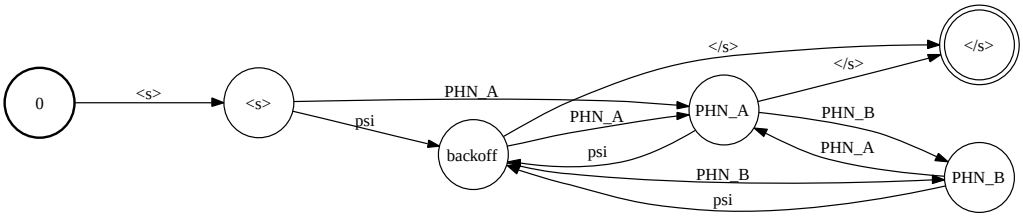

Figure 5.2: Example of a subword (phone) language model.

WFST $G_{subword}$. The hybrid "language model" is created by composition of word and subword language models

$$G_{subword} \circ G_{word}. \tag{5.2}$$

The substitution is illustrated in figure 5.3. The red part of network is the <unk> in figure 5.1 substituted by the subword model in figure 5.2.

The word dictionary L mapping words to phones is joint with the subword dictionary mapping subword labels to phones. Then this dictionary is converted to WFST representing the hybrid lexicon. Modified composition of the hybrid recognition network is written as:

$$H \circ C \circ (L_{word} \cup L_{subword}) \circ G_{subword} \circ G_{word}, \tag{5.3}$$

where H represents the HMM (tied-list) and C represents the mapping from context-dependent to context-independent phonetic units, L_{word} is the pronunciation dictionary mapping phones to words, $L_{subword}$ maps phones to subword units (eg. syllables, multigrams or phones). $G_{subword}$ is a weighted transducer created from the subword language model and G_{word} represents the word language model (weighted acceptor).

The <unk> and <silsp> nodes in the hybrid network (figure 5.3) produce an output label. The <unk> node produces symbol <unk> which is used as a marker of the beginning of subword section in the output. The <silsp> node produces symbol <silsp> which is used as a marker of the end of subword section in the output and also represents a *sil*/*sp* model[1].

Parameters such as *word insertion penalty* and *acoustic or language model scaling factors* are tuned to control the recognition accuracy and output of the LVCSR system. However, the hybrid network is considered as one object by the *SVite* decoder. The same penalty and scaling factor apply to both word and subword parts. That is why three different parameters are incorporated into the combined network during its building. The first parameter is **subword language model scaling factor**

[1]The *silence*/*short pause* is attached to each word model by default in each word network. This is used for modeling of possible silences following the words.

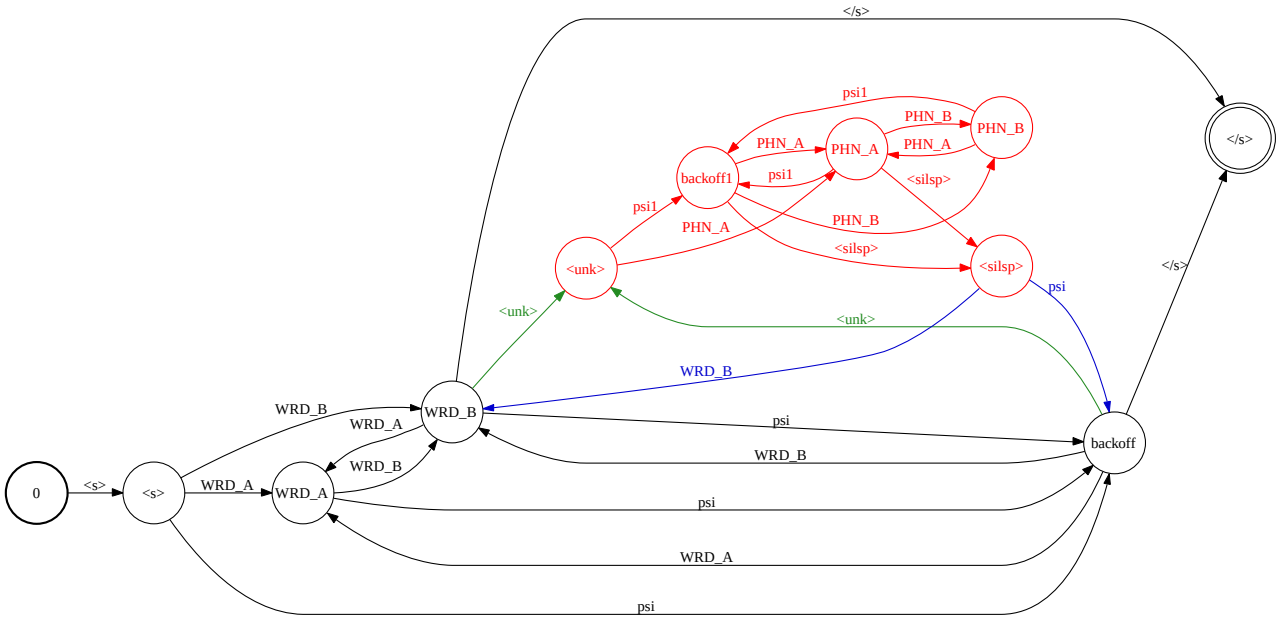

Figure 5.3: Example of hybrid word-subword language model.

SLMSF. This parameter exponentiates the likelihood assigned to the subword LM transitions. The second parameter is **subword word insertion penalty** *SWIP*. It is a constant which multiplies each transition's likelihood value leading to a word node. The last parameter is **subword cost** *SC*. It is a constant which multiplies the <unk> symbol likelihood and represents a cost of going to the whole subword model.

5.2 Hybrid recognition using multigrams trained on hand-made LVCSR dictionary

Multigrams trained on the LVCSR dictionary are used in [1]. We did the same experiment for better comparison. The WRDRED pronunciation dictionary is taken and multigrams (maximal multigram length $l_{mgram} = 5$, multigram pruning $c_0 = 5$) are trained on the word pronunciations. Hybrid system using the WRDRED dictionary trained multigrams is denoted as **HybridMgramDictLVCSR**. The advantage of WRDRED dictionary is in its correctness, the pronunciations are carefully hand-checked.

We also process the WRDRED dictionary word labels by the *G2P* system. Hybrid system using this subword model (denoted as **HybridMgramDictG2P**) evaluates the influence of *G2P* conversion accuracy on the word or STD accuracy. Comparison of these two systems is in figures 5.4 and 5.5.

We conclude, that the *G2P* conversion has no significant negative influence on the accuracy. UBTWV-IV is influenced slightly negatively, on the other hand the

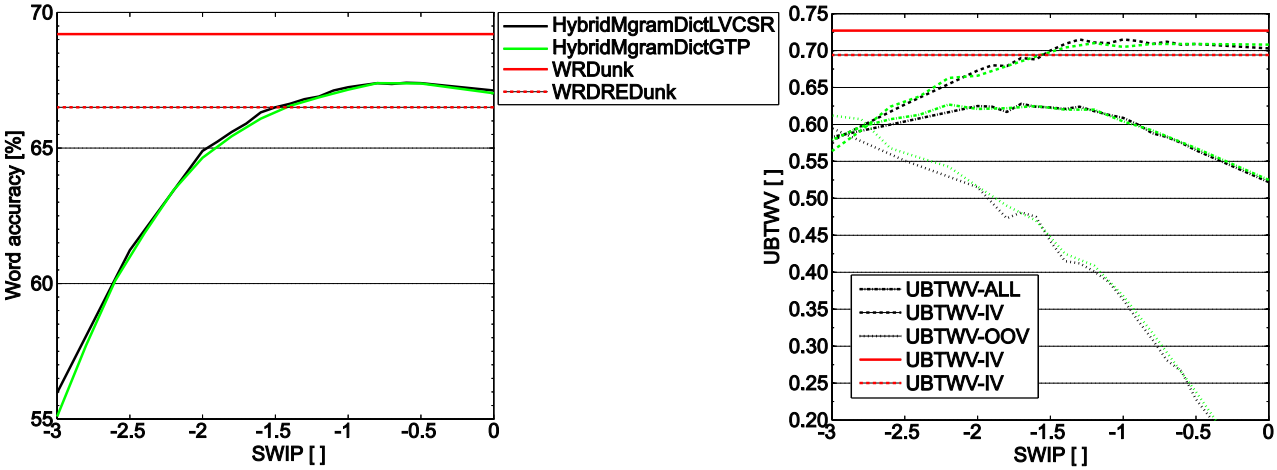

Figure 5.4: Dependency of the HybridMgram systems WAC and UBTWV on the parameter *SWIP*. The red color denotes the baseline systems WRDunk and WRDREDunk. Pronunciations of words of HybridMgramDictLVCSR system are taken from the WRD dictionary. Pronunciations of words of HybridMgramDictG2P system are generated automatically by the *G2P* tool.

Mgram UBTWV-OOV is influenced positively for certain values of *SWIP* parameter. The HybridMgramDictLVCSR will be used in several further experiments because we want to be comparable to Bazzi [1].

5.3 Memory and speed

Memory consumption and system speed (decoding time) are important factors for practical use. We evaluate the **real-time factor**[2] – *RT factor* and memory allocated by the decoder after loading the recognition network and acoustic model. Real-time factors are measured without feature extraction which has a constant RT factor and is the same in all experiments. Also, time consumed by spoken term detection algorithm is not included into RT factor, because it represents only fraction of the time. Both RT factor and allocated memory depend on the implementation of the decoder, so they can vary for different decoders. Decoding speed is tested on Intel® Xeon® CPU, model E5345 at frequency 2.33GHz processor with sufficient size of RAM.

Table 5.1 compares hybrid systems to the baseline systems from memory and index size, accuracy and speed points of view. The first part of the table compares baseline systems with fixed beam pruning $Pr = 220$. We see that both multigram systems (xwrd and noxwrd) are significantly (up to 8 times) slower and produce significantly larger indexes (up to 18 times) compared to WRDunk. On the other

[2]Proportion of the time of 1 CPU core needed to decode a portion of acoustic data to the time length of the portion of acoustic data.

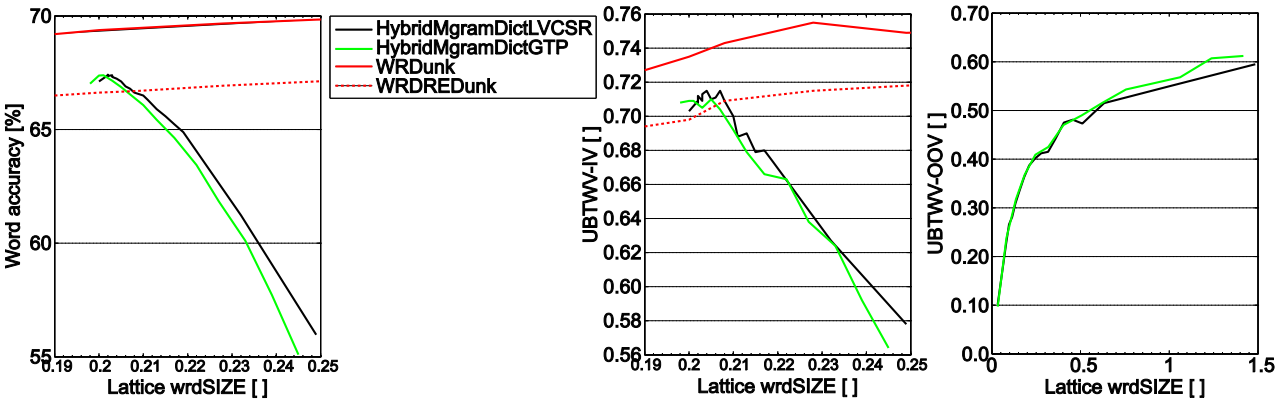

Figure 5.5: Dependency of the HybridMgram systems WAC, UBTWV and lattice size on the parameter *SWIP* is tuned. The red color denotes the baseline systems WRDunk and WRDREDunk. Pronunciations of words of HybridMgramDictLVCSR system are taken from the WRD dictionary. Pronunciations of words of HybridMgramDictG2P system are generated automatically by the *G2P* tool.

hand, the multigram xwrd system with unigram LM consumes only tenth of RAM compared to the other systems. The multigram xwrd system was chosen as the baseline system of OOV terms detection because it reaches the highest UBTWV-OOV accuracy.

The baseline hybrid systems (HybridMgramDictLarge and HybridMgramDictLarge2gr) with beam pruning $Pr = 200$ run relatively fast ($RT = 2$) and provide UBTWV-IV close to the WRDREDunk system. The UBTWV-OOV accuracy is about the half of xwrd system, but the index size is one tenth.

A **combined baseline system** is combination of baseline systems (WRDREDunk and xwrd) after the decoding, on the search level. "Combined" UBTWV-IV accuracy is the accuracy of WRDREDunk system, UBTWV-OOV accuracy is the accuracy of xwrd system. RT factor and index size are sums of word and sub-word systems.

Bottom three parts of table 5.1 present more precise comparison of hybrid and combined baseline system. We took 3 hybrid systems (HybridMgramDictLarge2gr, HybridMgramDictLarge and HybridMgramDictLVCSR2gr) and tuned the combined baseline systems to produce comparable UBTWV accuracy. Then these systems can be compared from speed and index size point of view.

The first system is the "star" system from the previous section. It is HybridMgramDictLarge2gr system with decoder beam pruning $Pr = 350$, subword scaling parameters $SLMSF = 1.0$, $SWIP = -0.8$ and $SC = 0.0$. This system was chosen as the system which produces the best UBTWV-OOV having UBTWV-IV and word index size comparable with the baseline WRDREDunk in the previous section. Here we see, that apart from the best accuracy, this system is not much usable

in practice. The real time factor is $RT = 28.4$ which is two time slower than the *combined baseline system* with real time factor sum $RT = 14$. One advantage of this system is the index size which is about 40% smaller.

HybridMgramDictLarge system with the same subword scaling parameters is chosen as the second hybrid system. The beam pruning $Pr = 250$ is set in this case. The real time factor $RT = 5.25$ of the combined baseline improves to $RT = 4.15$, which is 20% faster. Also, the hybrid index size is 38% smaller than the sum index size of the combined baseline system.

The last system is HybridMgramDictLVCSR2gr system with $Pr = 270$ and subword scaling parameters *SLMSF* $= 1.0$, *SWIP* $= -1.3$ and *SC* $= 0.0$. Compared to HybridMgramDictLarge2gr, this hybrid system with bigram subword language model consumes about 500MB less RAM. The UBTWV accuracy is close to saturated accuracies of the combined baseline system. If the baseline systems are tuned to produce comparable accuracy, this hybrid system is 9% faster and achieves only 34% of the index sizes of the baseline systems. The UBTWV-OOV accuracy deterioration is 0.027 against the "reasonably" saturated xwrd multigram baseline 0.647. The UBTWV-IV accuracy deterioration is 0.031 against the "reasonably" saturated WRDREDunk baseline 0.754.

The analysis of CPU and memory or disk consumptions of hybrid systems shows that hybrid systems are faster and reduce needed disk space for storage of the indexes. However, a hybrid system tuned to achieve accuracy comparable to "saturated" combined baseline system is about two times slower. If the accuracy is not the most important quality of STD system, hybrid system can provide very good performance. The needed decoding time can be reduced by 10% and the index size by 66% at the cost of 5% deterioration of UBTWV accuracy (HybridMgramDictLVCSR2gr system).

We conclude, that hybrid system based on LVCSR vocabulary (HybridMgramDictLVCSR2gr) is faster and consumes less memory than the HybridMgramDictLarge2gr system. This is caused by about two times smaller subword part. Also, the decoder can influence the RT factor. We noticed that the RT of HybridMgramDictLarge2gr system for higher beam pruning increased significantly more than linearly. This was not observed in case of WRDREDunk system.

System	LM order	# WRDn-grams		# SWRD n-grams		RAM	Pr	RT	UBTWV			wrdSIZE		
		1	2	1	2 (3)				ALL	IV	OOV	sum	wrd	swrd
WRDunk	2	50.0k	2.0M	–	–	550MiB	220	1.36	0.724	0.727	0.715	0.190	0.190	–
WRDREDunk	2	49.2k	1.6M	–	–	470MiB	220	1.32	0.486	0.694	–	0.190	0.190	–
xwrd	1	–	–	3.0k	–	22MiB	220	7.11	0.537	0.492	0.642	3.540	–	3.540
noxwrd	3	–	–	3.0k	451k (161k)	315MiB	220	10.75	0.630	0.647	0.593	1.740	–	1.740
HybridMgramDictLarge	$2+1$	49.2k	1.6M	7.8k	–	685MiB	220	2.00	0.592	0.708	0.320	0.335	0.200	0.135
HybridMgramDictLarge2gr	$2+2$	49.2k	1.6M	7.8k	136.8k	2310MiB	220	2.40	0.608	0.701	0.391	0.337	0.198	0.138
WRDREDunk	2	49.2k	1.6M	–	–	470MiB	330	7.79	0.528	0.754	–	0.950	0.950	–
xwrd	1	–	–	3.0k	–	22MiB	210	6.19	0.528	0.489	0.619	2.960	–	2.960
HybridMgramDictLarge2gr	$2+2$	49.2k	1.6M	7.8k	136.8k	2310MiB	350	28.37	0.713	0.753	0.620	2.400	1.000	1.400
WRDREDunk	2	49.2k	1.6M	–	–	470MiB	226	1.67	0.499	0.714	–	0.210	0.210	–
xwrd	1	–	–	3.0k	–	22MiB	160	3.58	0.481	0.470	0.500	1.400	–	1.400
HybridMgramDictLarge	$2+1$	49.2k	1.6M	7.8k	–	685MiB	250	4.15	0.651	0.715	0.501	1.000	0.340	0.660
WRDREDunk	2	49.2k	1.6M	–	–	470MiB	260	3.24	0.512	0.723	–	0.380	0.380	–
xwrd	1	–	–	3.0k	–	22MiB	210	6.19	0.528	0.489	0.619	2.960	–	2.960
HybridMgramDictLVCSR2gr	$2+2$	49.2k	1.6M	4.0k	42.3k	1855MiB	270	8.62	**0.691**	**0.723**	**0.615**	1.100	0.440	0.700

Table 5.1: Comparison of memory and CPU requirements of hybrid systems. Column *order* denotes the order of used language models. $2+1$ means bigram word and unigram subword LM. The following 4 columns contain the numbers of particular unigrams/bigrams. The number in brackets is the number of trigrams for the noxwrd system. Occupied memory after the recognition network is loaded by the decoder is in column *RAM*. Acoustic model is not included. Its size is constant for all experiments: 190MB. *Pr* denotes chosen beam pruning. RT is the estimated real time factor. Columns UBTWV and wrdSIZE denote the accuracies and index sizes of particular systems. The first part of the table (the first 6 rows) compares baseline and hybrid systems having beam pruning $Pr = 220$. The following 3 parts of the table show three different hybrid systems and appropriate baseline systems having comparable UBTWV accuracy. The real time factor is estimated on Intel® Xeon® CPU, at frequency 2.33GHz.

5.4 Conclusion

Conclusions of hybrid word-subword systems are given in this section. Theoretically, we should obtain better IV and OOV accuracy with equal or smaller lattice size (index size) with the hybrid system than is achieved by the combination of standalone systems at the level of term detection. Our experiments have however shown, that this can be achieved only for a certain range of system parameter settings (beam pruning, hybrid network penalties and scales). If standalone systems are tuned to the best accuracy (separately), this combination is not overcome by the hybrid system presented in this thesis. We can only make it smaller and faster, but still with little deterioration of accuracy.

It is interesting to notice, that the UBTWV-OOV of pure subword multigram system (xwrd) is not outperformed by the hybrid systems, while for UBTWV-IV the hybrid systems show superiority. This can be explained by the inaccuracy of estimated place of out-of-vocabulary words (represented by <unk>).

Our explanation, why UBTWV-OOV is not better for the hybrid than for pure subword multigram system, is the following: the strong word-model in the hybrid system can cause that for an OOV, the sub-word model is not activated at all – the system does not enter the <unk> part of the recognition network. In this case, the system actually misses the OOV without any chance to recover it, resulting in a miss. This problem does not occur in subword systems where the whole utterance is recognized in subwords (so no misses are produced).

On the other hand, a false alarm of OOV occurrence does not necessarily cause that the IV term (overlapped with the OOV false alarm) is not detected. In this case, the IV can be still converted to subwords and searched in the subword form.

This leads to conclusion, that it is important not to miss the OOV parts of utterances for hybrid systems. The position of <unk> must be estimated accurately.

Chapter 6

Conclusion and discussions

The thesis deals with spoken term detection. The corner stone of this thesis is search of out-of-vocabulary terms which are not present in dictionary of word-based speech recognizer. We investigate into combination of word and subword approaches to get the best search accuracy (especially for out-of-vocabulary words) having the highest search speed and the lowest memory consumptions.

Several systems were described and tested in this thesis. We aimed at evaluation of spoken term detection accuracy. The accuracy was evaluated on 3h of conversational telephone speech. We searched for nearly 400 terms (having up to 4 words) where about one third contains at least one out-of-vocabulary word.

The Upper-Bound-Term-Weighted-Value (UBTWV) was used as the primary evaluation metric. We derived this metric from Term-Weighted-Value (TWV) defined by NIST. The difference is in calibration of terms' scores to one global threshold. The UBTWV shifts the terms confidences to maximize term's TWV for threshold 0. Terms are then pooled and average upper-bound TWV is calculated. By this, we can effectively bypass the calibration of scores and concentrate on the actual system's accuracy.

We also evaluated word accuracy and size of output produced by systems. The size of the output is important from the practical point of view.

The baseline LVCSR system was used to demonstrate the effect of missing words in the vocabulary. We have shown a deterioration of spoken term detection and word recognition accuracy. The baseline recognizer was also used to demonstrate tuning the recognizer to "reasonably best" accuracy.

One of set of subword systems were phone multigram systems. We adopted the approach of phone multigrams published by Deligne and Bimbot [5]. First, we found the optimal configuration of phone multigram model according to spoken term detection accuracy. We proposed two new multigram models with constraints (nosil and noxwrd). These constrained phone multigrams were found superior to the

baseline multigrams. Beside the evaluation of term confidences, we evaluated also the influence of number of out-of-vocabulary term segmentations to multigrams. It was found, that this number of segmentations has significant impact only on the accuracy of out-of-vocabulary term detection. The conclusion is that constrained multigrams significantly overcome standard multigrams and phones. This system is more accurate, faster and produces smaller lattices.

The last set were hybrid word-subword systems. A framework based on WFST was defined for construction of hybrid word-subword language models. We investigated also the dependency of size of lattices and accuracy on parameters of hybrid language model. A hybrid system can achieve higher accuracy than a word system having comparable size of produced lattices.

We trained phone multigram model only on pronunciation vocabulary. The baseline hybrid system was based on pronunciation multigrams derived from LVCSR dictionary (similarly as proposed by Bazzi [1]). We extended this baseline system further by training the multigram model on large dictionary of out-of-vocabulary words. This system achieved slightly better accuracy. We tested the influence of automatic grapheme-to-phoneme production of pronunciations of out-of-vocabularies on the spoken term detection accuracy. The influence was not significant.

The second extension was incorporation of bigram language model over multigrams in the subword part of the hybrid recognizer. The effect of stronger subword model becomes evident on the accuracy of out-of-vocabulary terms and smaller lattice size.

The hybrid system was evaluated from the lattice size and computational speed points of view. This should ensure practical applicability of the proposed system. We found that hybrid system can achieve slightly worse accuracy with significant reduction of lattice size and the same speed compared to the combined word and multigram systems. From pure accuracy point of view, this could be considered a failure of the proposed approach, but in our opinion, this drawback is largely compensated by its simplicity and efficiency – keep in mind that in the combination of word and multigram systems, the data must be processed separately by both systems which is much more complicated.

6.1 Future work

The main problem of our hybrid approach is in the requirement of correct scaling of word and subword language models. In the case of incorrectly scaled language models, produced lattices can contain too high or too low number of subword units.

The accuracy will not be optimal.

An approach published by Yazgan and Saraclar [16] based on estimation of hybrid language model on hybrid textual corpora could be more robust. There is no need of any scaling of word-subword parts of the LM. The "scaling" parameters are directly estimated on the data. On the other hand, our approach can be used in cases where we do not have much LM training data, or we want to adapt the system to a different domain.

We could build an LM containing several different types of OOV symbol as `<unk-name>`, `<unk-street>`, `<unk-city>` and `<unk-other>` for each domain. Then, each domain-dependent subword language model could be estimated separately. `<unk-name>` on names of people, `<unk-street>` on names of streets, etc. In a new scenario, where the set of names radically differs from the set used for training of the previous `<unk-name>` model, only the new `<unk-name>` would need to be be trained and hybrid network recompiled. As the corpora for the subword language models are in "dictionary" format, building them is easy. The approach based on hybrid textual corpora can not be so easily adapted to the new set of names. The names appearing in the hybrid corpora would need to be substituted by the new names and the whole hybrid language model would need to be retrained.

This work could be also highly beneficial for applications, where adding new words to the system is requested. Having the `<unk-name>`, `<unk-street>`, `<unk-city>`, `<unk-other>` etc. symbols in the recognition network would actually create the necessary place-holders, where appropriate new words could be added off- or even on-line.

It would be also very interesting to evaluate the phone multigram model proposed by Bisany and Ney [3] and compare it to our approach. Their multigram model is defined theoretically more correctly and should achieve better results. It would be also interesting to assess the improvement in the hybrid system and in a standalone multigram system.

In the proposed constrained multigrams, we could also improve dealing with the silence, that is currently considered as an independent unit. The silence models (`sil` and `sp`) could be placed to the end of each multigram unit in the same way as it is done in LVCSR.

Another future work is linked with using the hybrid system as OOV detector and comparing its performances with the detector based on strong vs. weak posteriors, as it was proposed at JHU workshop in 2007[1] by Hermansky et al. [11]. Efforts in this direction are already running within the European DIRAC project [4].

[1] `http://www.clsp.jhu.edu/ws2007/groups/rmimsr/`

Bibliography

[1] I. Bazzi. *Modelling Out-of-vocabulary Words for Robust Speech Recognition*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science., 2002.

[2] M. Bisani and H. Ney. Open vocabulary speech recognition with flat hybrid models. In *Proceedings of Interspeech*, pages 725–728, Lisbon, Portugal, September 2005.

[3] M. Bisani and H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008.

[4] J. Černocký and H. Hermansky. Report on identification of repeatedly occurring out-of-vocabulary words, deliverable no: D2.12, DIRAC project. Technical report, Brno University of Technology, Czech Republic, 2010.

[5] S. Deligne and F. Bimbot. Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams. In *Proceedings of ICASSP*, pages 169–172, Detroit, MI, USA, 1995.

[6] J. Fiscus, J. Ajot, and G. Doddington. The spoken term detection (STD) 2006 evaluation plan. Technical report, National Institute of Standards and Technology (NIST) USA, September 2006.

[7] J. Fiscus, J. Ajot, J. Garofolo, and G. Doddington. Results of the 2006 spoken term detection evaluation. In *The 2007 Special Interest Group on Information Retrieval (SIGIR-07) Workshop in Searching Spontaneous Conversational Speech*, July 2007.

[8] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký. Probabilistic and bottle-neck features for LVCSR of meetings. In *Proceedings of ICASSP*, pages 757–760. IEEE Signal Processing Society, 2007.

[9] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiát, and V. Wan. The AMI Meeting Transcription System. In *Proceedings of NIST Rich Transcription*

2006 Spring Meeting Recognition Evaluation Worskhop, page 12. National Institute of Standards and Technology, 2006.

[10] T. Hain, V. Wan, L. Burget, M. Karafiát, J. Dines, J. Vepa, G. Garau, and M. Lincoln. The ami system for the transcription of speech in meetings. In *Proceedings of ICASSP*, pages 357–360. IEEE Signal Processing Society, 2007.

[11] H. Hermansky, L. Burget, P. Schwarz, P. Matějka, M. Hannemann, A. Rastrow, C. White, S. Khudanpur, and J. Černocký. Recovery from model inconsistency in multilingual speech recognition, report from JHU workshop 2007. Technical report, Johns Hopkins University, USA, 2007.

[12] A. Le. NIST's workshop presentation on STT, NIST RT 2004, November 2004.

[13] K. Ng. *Subword-Based Approaches for Spoken Document Retrieval*. PhD thesis, Massachusetts Institute of Technology, USA, February 2000.

[14] P. Schwarz, P. Matějka, and J. Černocký. Towards lower error rates in phoneme recognition. *Lecture Notes in Computer Science*, 2004(3206):465–472, 2004.

[15] I Szöke, M. Fapšo, M. Karafiát L., Burget, F. Grézl, P. Schwarz, O. Glembek, P. Matějka, J. Kopecký, and J. Černocký. Spoken term detection system based on combination of LVCSR and phonetic search. *Lecture Notes in Computer Science*, 2008(4892):237–247, 2008.

[16] A. Yazgan and M. Saraclar. Hybrid language models for out of vocabulary word detection in large vocabulary conversational speech recognition. In *Proceedings of ICASSP*, volume 1, pages 745–748, May 2004.

Author

Ing. Igor Szöke
http://www.fit.vutbr.cz/~szoke

Igor was born on July 26, 1980 in Brno, Czech Republic. He received his Master's degree in Information Technology from the Brno University of Technology in June 2003. The topic of his diploma project was the Czech text-to-speech system using Harmonic plus Noise model.

He joined the Speech@FIT group at Brno University of Technology as a Ph.D. student in September 2003. There he started to work on very low bitrate coding. He was on an internship at ESIEE Paris in France from February to July 2004. Under a guidance of Prof. Genevieve Baudoin, he worked on automatically derived speech units for very low bitrate coding.

After the return, he subsequently worked on M4, AMI and AMIDA projects on the topic of keyword spotting and spoken term detection. He participated as the team leader of Speech@FIT group at NIST STD 2006 evaluations.

During his graduate studies, Igor has authored and co-authored several conference papers presented on international events. He taught lectures and exercises in signal and speech related courses and supervized several Bc. and Ms. thesis. He is member of the IEEE and ISCA.