Hybrid word-subword spoken term detection

Igor Szöke*
Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic
szoke@fit.vutbr.cz

Abstract

The thesis investigates into keyword spotting and spoken term detection (STD), that are considered as sub-sets of spoken document retrieval. It deals with two-phase approaches where speech is first processed by speech recognizer, and the search for queries is performed in the output of this recognizer. Standard large vocabulary continuous speech recognizer (LVCSR) with fixed vocabulary is not capable incapability of detecting out-of-vocabulary words (OOV). A hybrid spoken term detection system combining both word and subword parts in one recognition network is proposed. Extensive experiments investigating into different variants of this approach are performed, and the results (in terms of spoken term detection precision, speed, and necessary computing resources) are reported on standard data from NIST STD 2006 evaluation.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]; H.5.1 [**Multimedia Information Systems**]

Keywords

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

1. Introduction

The research field of this thesis is spoken term detection. The corner stone of this thesis is dealing with 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.

*Recommended by thesis supervisor: Dr. Jan Černocký
Defended at Faculty of Information Technology, Brno University of Technology on December 13, 2010.

Szöke, I. Hybrid word-subword spoken term detection. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 2, No. 2 (2010) 121-129

Short definition of important terms is placed in the following paragraph to avoid confusion of the reader of this thesis.

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** (**STD**) context. 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.

Spoken term detection system is a system for spotting (searching) given terms in given speech data. On contrary to a *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 STD system takes a set of terms and output of a speech recognizer and produces a list of putative hits of given term. Our spoken term detector is based on a **large vocabulary continuous speech recognizer** – LVCSR. The speech recognizer is mainly taken "as is" and is described in section 4. 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 of more complex queries.

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. 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 closed set of words to be recognized and word language model prefers likely word sequences off the "exotic" ones (probably carrying more information). Also, it is known that if an OOV appears, it usually causes no 1 word error, but approxi-

mately 2 – 4 word errors [2]. This is a justification of our investigation into subword recognition.

2. Spoken term detection

The generic scheme of a spoken term detection system is in figure 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 2) which contain speech transcribed in words labels. The lattices are searched for the given terms or keywords.

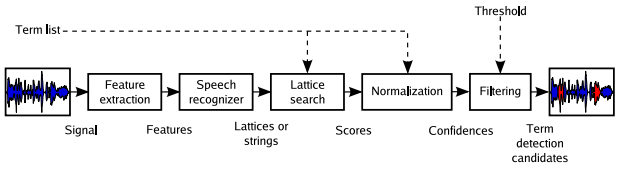

Figure 1: General scheme of spoken term detector.

Spoken term detection 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 the 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 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 disadvantage of this approach is that 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).

2.1 Theoretical background

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 contains the term in given time, and the best sequence is found:

$$\hat{\mathbf{w}}(term_{t_b}^{t_e}) = \underset{\mathbf{w}(term_{t_b}^{t_e}) \in \mathcal{W}(term_{t_b}^{t_e})}{\arg\max} p(\mathbf{w}(term_{t_b}^{t_e})|\mathcal{D}), \quad (1)$$

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

$$\hat{\mathbf{w}}(term_{t_b}^{t_e}) = \underset{\mathbf{w}(term_{t_b}^{t_e}) \in \mathcal{W}(term_{t_b}^{t_e})}{\arg\max} \frac{p(\mathcal{D}|\mathbf{w}(term_{t_b}^{t_e}))p(\mathbf{w}(term_{t_b}^{t_e}))}{\sum_{\mathbf{w}' \in \mathcal{W}} p(\mathcal{D}|\mathbf{w}')p(\mathbf{w}')}. \quad (2)$$

In practice, direct implementation of formula 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 a 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:

$$p(term_{t_b}^{t_e}) = \frac{p(\mathcal{D}|\mathbf{w}(term_{t_b}^{t_e}))p(\mathbf{w}(term_{t_b}^{t_e}))}{\sum_{\mathbf{w}' \in \mathcal{W}'} p(\mathcal{D}|\mathbf{w}')p(\mathbf{w}')}. \quad (3)$$

2.2 Search in lattice

This section presents the "implementation" of the calculation of term posterior probability stated in equation 3 in the previous section. Lattices (figure 2) are nowadays used as the multiple hypothesis output of speech recognizer.

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 $u = U(a)$ 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 N is a large number.

Searching for the term in the lattice is more robust than searching for the term in the string output (1-best). Having the lattice, we have the $\mathcal{W}'$ and we can estimate the posterior probability of term according to equation 3. The posterior probability gives confidence of term for particular occurrence of term (represented by arc a) in time $t_b(a), t_e(a)$.

3. Evaluation

Conversational Telephone Speech (CTS) data from 2006 NIST Spoken Term Detection evaluations (NIST STD06) [4] were used in our experiments. For our tests, they are however not representative as the original NIST STD06 development term set for CTS contains low number of OOVs. Therefore, first of all, all terms containing

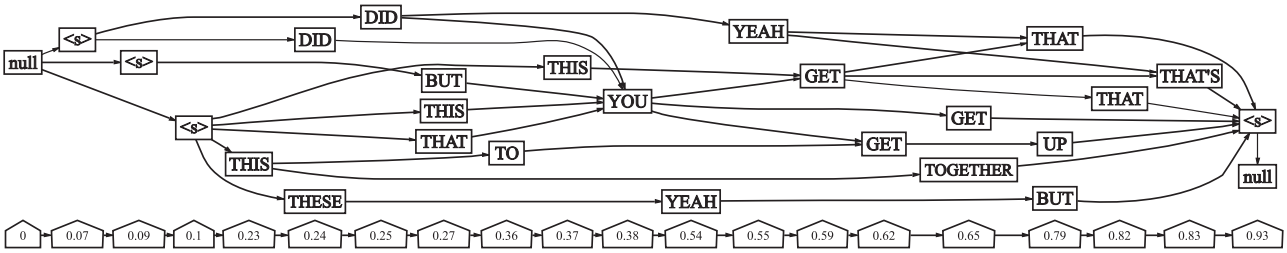

Figure 2: An example of word lattice. X-axis represents time.

true OOVs were omitted. Then, a set "artificial" OOV was defined. A limited LVCSR system was created (denoted by *WRDRED* which means "reduced vocabulary") where 880 words were omitted from the vocabulary. We selected 440 words from the term set and other 440 words from the LVCSR vocabulary. This system had 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 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*). System parameters (decoder insertion penalties and scaling factors) are tuned on the test set.

3.1 UBTWV - Upper Bound TWV

We used Term Weighted Value (TWV) for evaluation of spoken term detection (STD) accuracy of our experiments. The TVW was defined by NIST for STD2006 evaluation [4]

$$TWV(thr) = 1 - \underset{term}{average}\{p_{MISS}(term, thr) + \beta p_{FA}(term, thr)\}, \quad (4)$$

where β is 999.9. The $p_{MISS}(term, thr)$ is miss probability of the $term$ and given threshold thr. The term false alarm probability is denoted $p_{FA}(term, thr)$.

One drawback of TWV metric is its one global threshold for all terms. This is good for evaluation for end-user environment, but leads to uncertainty in comparison of different experimental setups, as we do not know if the difference is caused by different systems or different normalization and global threshold estimation. This is a reason for *Upper Bound TWV* (UBTWV) definition which differs from TWV in individual threshold for each term. The ideal threshold for each term is found to maximize the term's TWV:

$$thr_{ideal}(term) = \arg\max_{thr} TWV(term, thr) \quad (5)$$

The UBTWV is then defined as

$$UBTWV = 1 - \underset{term}{avg}\{p_{MISS}(term, thr_{ideal}(term)) + \beta p_{FA}(term, thr_{ideal}(term))\}. \quad (6)$$

It is equivalent a shift of each term to have the maximal $TWV(term)$ at threshold 0. Two systems can be compared by UBTWV without any influence of normalization and ideal threshold level estimation on the systems TWV score. The *UBTWV* was evaluated for the whole set of terms (denoted *UBTWV-ALL*), only for in-vocabulary subset (denoted *UBTWV-IV*) and only for out-of-vocabulary subset (denoted *UBTWV-OOV*).

4. Word recognition

This section describes the recognizer used for experiments stated in the thesis. During the pre-processing, the acoustic data was split into shorter segments in silences (output of speech/nonspeech detector) longer than 0.5s. The data was also split if the speaker changed (based on the output of diarization). Segments longer than 1 minute were split into 2 parts in silence the closest to the center of the segment. This was done to overcome long segments and accompanying problems during decoding.

Acoustic models from an LVCSR system were used for subword recognition. Presented LVCSR [6] is a state-of-the-art system derived from AMIDA LVCSR [7]. The system uses standard cross-word tied states triphone models and works in three passes of recognition (figure 3).

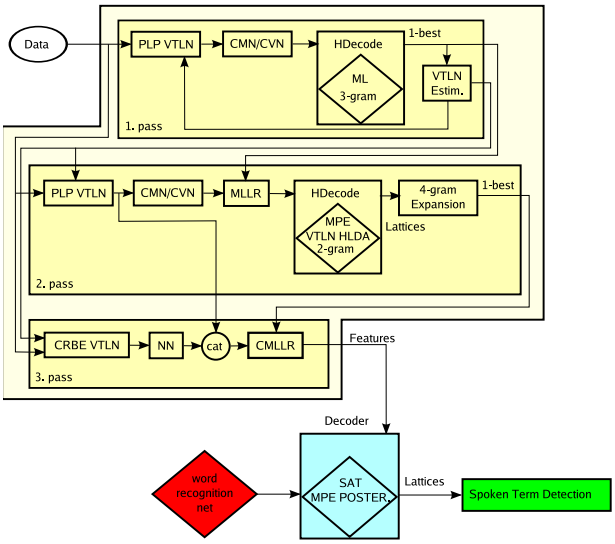

Figure 3: Schema of 3-pass recognition system used in this thesis. The system is derived from AMI LVCSR.

The acoustic models are trained on *ctstrain04* corpora which is a subset of *h5train03* set defined at the University of Cambridge [5]. Total amount of data is 277 hours. A bigram word language model (used for comparison) was trained on 977M words of a mix of 9 corpora. The corpora contain mainly conversation speech and round table meeting transcripts.

The same *ctstrain04* corpora was used as base phone corpora for our experiments (phone multigrams). The size is 11.5M phones.

Baseline word recognition systems were evaluated in ta-

ble 1. The baseline LVCSR system was denoted as *WRD* and reduced vocabulary (880 words were omitted from the vocabulary) LVCSR was denoted as *WRDRED*.

System	WAC	Word UBTWV		
		ALL	IV	OOV
WRD	69.20	0.724	0.727	0.715
WRDRED	66.50	0.486	0.694	0.000

Table 1: Comparison of word accuracy (WAC) and UBTWV of different baseline LVCSR systems. WRD is the "full" vocabulary baseline, WRDRED is the reduced vocabulary baseline.

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, 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 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.

5.1 Building combined word-subword hybrid recognition network

We used our static decoder *SVite* for hybrid recognition experiments. The only one modification was realized in the **network** for hybrid recognition/decoding.

The network can be seen as a weighted finite state transducer (*WFST*) which maps a sequence of HMM models to a sequence of word labels which are accepted by a language model (weighted finite state acceptor).

The WFST is a finite state device that encodes a mapping between input and output symbol sequences. A weighted transducer associates weights such as probabilities, durations, penalties or any other quantity that accumulates linearly along paths, to each pair of input and output symbol sequence. WFST provide a natural representation of HMM models, pronunciation dictionary and language model [8]. Weighted determinization and minimization algorithms optimize their time and space requirements, and a weight pushing algorithm distributes the weights along the paths of a weighted transducer optimally for speech recognition.

Consider a pronunciation lexicon and take its Kleene closure by connecting an ϵ-transition from each final state to the initial state. The resulting pronunciation lexicon can transcribe any sequence of words from the vocabulary to the corresponding phoneme sequence.

Consider a language model G and a pronunciation lexicon L. The composition of these two WFSTs,

$$L \circ G, \tag{7}$$

gives a transducer that maps from phones to word sequences while assigning a language model score to each such sequence of words. Incorporating context-dependent triphone models is a simple matter of composing

$$C \circ L \circ G, \tag{8}$$

where C represents the mapping from context-dependent to context-independent phonetic units. Then, incorporating HMM models H:

$$H \circ C \circ L \circ G, \tag{9}$$

results to the transducer capable of mapping distributions to word sequences restricted to the language model G. The hybrid word-subword recognition network can be built by

$$H \circ C \circ (L_{word} \cup L_{subword}) \circ G_{subword} \circ G_{word}, \tag{10}$$

where H and C are the same as in Eq. 9, 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 a word language model (weighted acceptor).

We used the OpenFST toolkit for building the recognition network.

5.2 Word model

We need the word and subword language models and dictionaries. The word LM must be open vocabulary, so it must contain an "`<unk>`" word. The "`<unk>`" is considered as the OOV word which will be modelled by the subword model (see Figure 4).

5.3 Subword model

The second input is a subword model. Simple phone bigram language model is shown as an example in the Figure 5. The `<unk>` symbol is replaced by this subword model.

The substitution is illustrated in the Figure 6. The red part of network is `<unk>` substituted by the subword model.

Parameters such as word insertion penalty and acoustic or language model scaling factors can be tuned to control the recognition accuracy and output of the LVCSR system. However, the hybrid network is considered as one unit by the decoder. The same penalty and scaling factor apply for both word and subword parts. That is why three different parameters were incorporated into the combined network during its building. The first parameter is subword language model scaling factor *SLMSF*. This parameter multiplies the log likelihood assigned to the subword LM transitions. The second parameter is the subword word insertion penalty *SWIP*. It is a constant which is add to each transition's log likelihood value leading to a

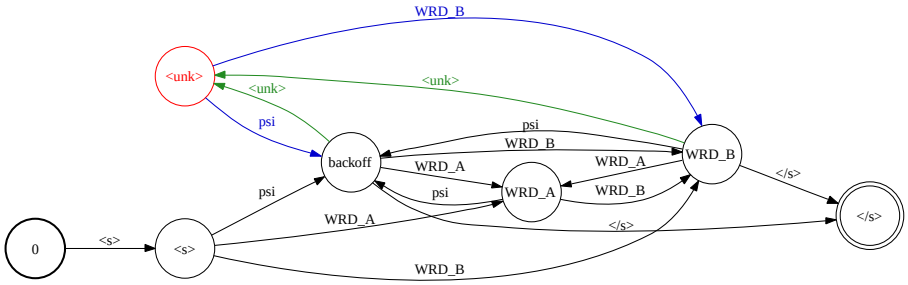

Figure 4: An example of open vocabulary language model. The <unk> states for the out-of-vocabulary words.

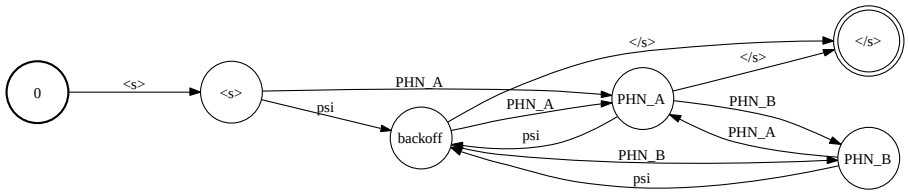

Figure 5: An example of a subword (phone) language model.

word node. The last parameter is a subword cost *SC*. It is a constant which is added to the **<unk>** symbol and represents a simple cost of going to the whole subword model.

5.4 Phone multigrams

Variable length sequences of phones are denoted as phone multigrams. The multigram language model was proposed by Deligne et al. [3]. 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:

$$L(\mathbf{w}, \mathbf{s}) = \prod_{i=1}^{q} p(s_i) \tag{11}$$

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

$$L_{mgr}^{best}(\mathbf{w}) = \max_{\mathbf{s} \in \mathcal{S}} L(\mathbf{w}, \mathbf{s}) \tag{12}$$

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:

$$\mathcal{P} = (p_i)_{i=1}^{m} \quad \text{where} \quad p_i = p(s_i) \quad \text{and} \quad \sum_{i=1}^{m} p_i = 1 \tag{13}$$

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:

$$\mathbf{s}^{*(k)} = \arg\max_{\mathbf{s} \in \mathcal{S}} L(\mathbf{s}|\mathbf{w}, \mathcal{P}^k) \tag{14}$$

According to [3], the re-estimation formula of i^{th} parameter (sequence) at iteration $k+1$ is intuitive:

$$\mathcal{P}_i^{k+1} = \frac{c(s_i, \mathbf{s}^{*(k)})}{c(\mathbf{s}^{*(k)})}, \tag{15}$$

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 over-training, 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. Multigram units which occur less than 5 time (multigram pruning factor) are omitted from the inventory.

Phone multigram system denoted as xwrd is used later in this paper. Desription of this system is out of the scope of this paper and was published in [9].

6. Hybrid recognition using multigrams trained on hand-made LVCSR dictionary

We used three different subword models. The first is a phone loop. The second and the third are phone multigram-based units. Because deep description of all systems and

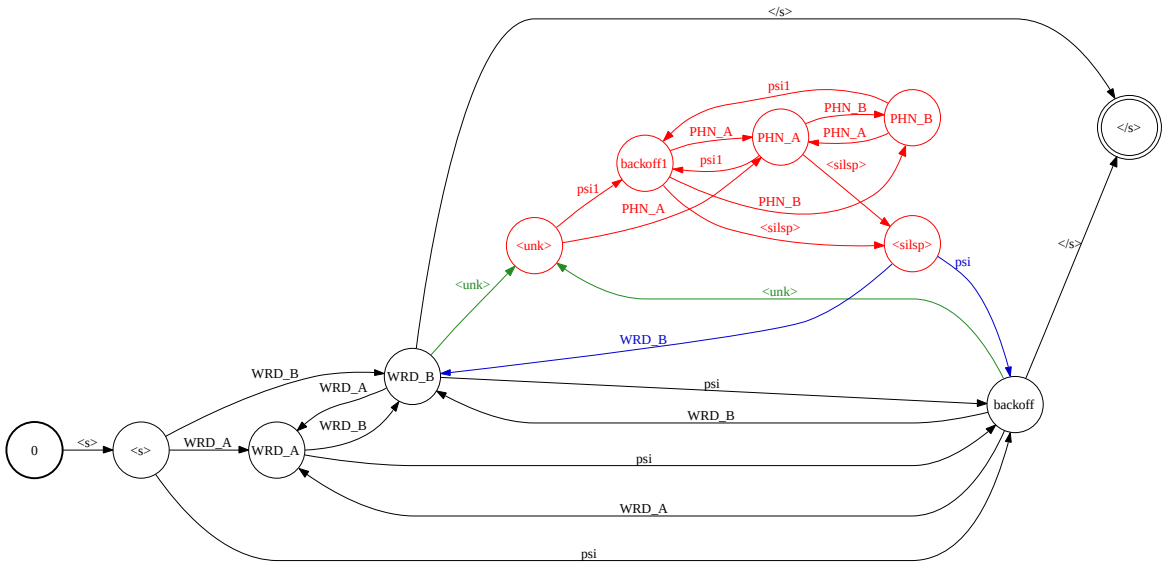

Figure 6: An example of hybrid word-phone language model where the <unk> symbol was substituted by the phone model.

experiments is out of the scope of this paper (the reader is refered to read the thesis), we selected only several important experiments.

The multigram training data are pronunciations of words in the following experiments. Each pronunciation variant is taken as one utterance (phone string). The language model built over the multigrams is also estimated on the segmented (by multigrams) pronunciation 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 (maximum 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.

System denoted as **HybridMgramDictLarge** is built using automatically generated pronunciation variants (by *G2P*[1]) of large number of words. A large set of word label is collected from the corpora used for the word language model estimation.

6.1 Hybrid systems based on manually versus automatically built dictionary

We compare the accuracies for hybrid system based on the LVCSR dictionary (HybridMgramDictLVCSR) and hybrid system based on the large dictionary (HybridMgramDictLarge). Both multigram systems are trained with multigram pruning parameter $c_0 = 5$.

Our cion is, that both systems are comparable. The HybridMgramDictLVCSR system is slightly better on UBTWV-IV and word accuracy, the HybridMgramDictLarge is slightly better on the Mgram UBTWV-OOV and achieves smaller size of the lattices for comparable word accuracy and UBTWV-IV. Figures 7 and 8 compare these systems.

6.2 Subword model with bigram language model

Our approach to build hybrid word-subword recognition networks allows to use subword language model with higher order than unigrams. We tested bigram subword language model. We compare both, the HybridMgramDictLVCSR and the HybridMgramDictLarge systems. The results are summarized in figures 9 and 10. From the word accuracy point of view, the HybridMgramDictLVCSR bigram system has slightly lower maximum accuracy than the unigram system. The HybridMgramDictLarge bigram system is slightly better than the unigram system on the word accuracy.

The UBTWV-ALL of bigram systems are comparable. Only UBTWV-OOV of HybridMgramDictLarge is significantly higher for certain range of *SWIP* parameter. UBTWV-IV of bigram systems are slightly lower for higher values of *SWIP* compared to the unigram systems. On the other hand, UBTWV-OOV are significantly higher. The greatest differences between unigram and bigram subword systems are seen on accuracy related to the same lattice size (figure 10). We can see, that the bigram systems have lattice sizes significantly reduced. The lattice size does not grow so fast with increasing accuracy while the *SWIP* is tuned.

Although HybridMgramDictLVCSR2gr and HybridMgramDictLarge2gr are comparable, we decide to select **HybridMgramDictLarge2gr** according to the higher Mgram UBTWV-OOV accuracy.

7. 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

[1] G2P: Automatic grapheme-to-phoneme conversion tool.

[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.

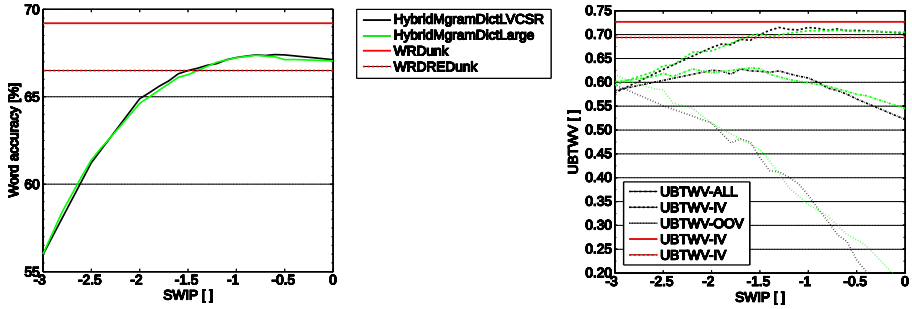

Figure 7: Dependency of the hybrid multigram systems WAC and UBTWV accuracy on the *SWIP* parameter for two different subword models. The HybridMgramDictLVCSR is trained on the hand made LVCSR WRDRED dictionary. The HybridMgramDictLarge is trained on dictionary created by *G2P* tool automatically trained on large word list derived from the word corpora. The dark gray denotes the baseline systems WRD and WRDRED. The multigram pruning parameter of trained multigrams is $c_0 = 5$ for both systems.

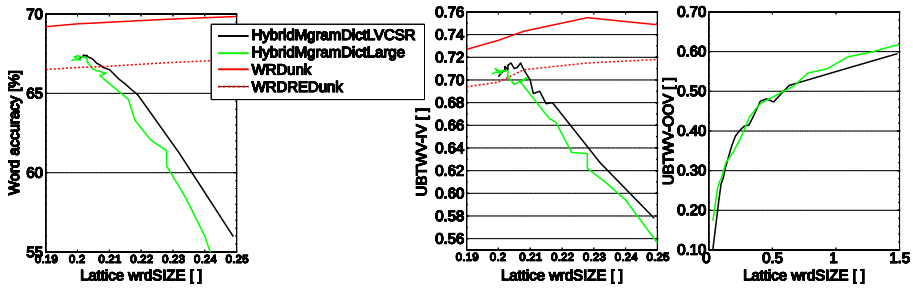

Figure 8: Dependency of the hybrid multigram systems WAC, UBTWV accuracy and lattice size on the *SWIP* parameter for two different subword models. The *HybridMgramDictLVCSR* is trained on the hand made LVCSR WRDRED dictionary. The HybridMgramDictLarge is trained on dictionary created by *G2P* tool automatically trained on large word list derived from the word corpora. The dark gray denotes the baseline systems WRD and WRDRED. The multigram pruning parameter of trained multigrams is $c_0 = 5$ for both systems.

the decoder after loading the recognition network and the 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 2 compares hybrid systems to the baseline ones from memory and lattice size (index size), accuracy and speed points of view. The first part of the table compares baseline systems. We see that xwrd multigram baseline system is significantly (up to 8 times) slower and produces significantly larger index (up to 18 times) compared to WRD. On the other hand, the multigram xwrd system with unigram LM consumes only one 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.

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

Selected hybrid HybridMgramDictLVCSR2gr system is compared to baseline systems. The UBTWV accuracy is close to saturated accuracies of the combined baseline system. If the baseline systems are tuned to produce comparable accuracy and speed, this hybrid system 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 0.642. The UBTWV-IV accuracy improvement is 0.029 against the "reasonably" saturated WRDRED baseline 0.694.

The analysis of CPU and memory or disk consumptions of hybrid systems shows that hybrid systems can be 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 (see the thesis). 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 sub-

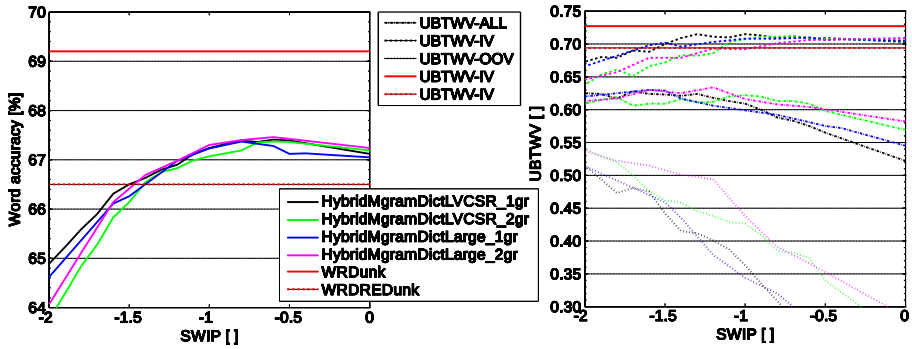

Figure 9: Dependency of the hybrid multigram systems word accuracy and UBTWV accuracy on the subword model. The HybridMgramDictLVCSR and the HybridMgramDictLarge systems are evaluated also with bigram language model in the subword part (*_2gr). The dark gray denotes the baseline systems WRD and WRDRED.

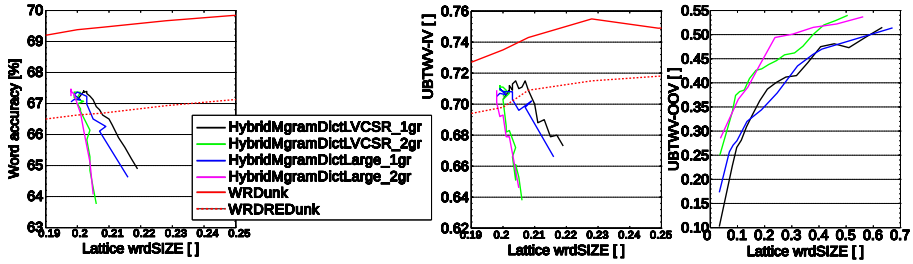

Figure 10: The dependency of the hybrid multigram systems word accuracy and UBTWV accuracy on the lattice size while parameter *SWIP* is tuned. The HybridMgramDictLVCSR and the HybridMgramDictLarge systems are evaluated also with bigram language model in the subword part (*_2gr). The dark gray denotes the baseline systems WRD and WRDRED.

word 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 WRDRED system.

8. Conclusion and discussions

The thesis deals with spoken term detection. The corner stone of this thesis is dealing with 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 [3]. 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 contribution 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

System	LM order	RAM	RT	UBTWV			wrdSIZE		
				ALL	IV	OOV	sum	wrd	swrd
WRD	2	550MiB	1.36	0.724	0.727	0.715	0.190	0.190	–
WRDRED	2	470MiB	1.32	0.486	0.694	–	0.190	0.190	–
xwrd	1	22MiB	7.11	0.537	0.492	0.642	3.540	–	3.540
HybridMgramDictLVCSR2gr	2 + 2	1855MiB	8.62	**0.691**	**0.723**	**0.615**	1.100	0.440	0.700

Table 2: Comparison of memory and CPU requirements of hybrid systems. Column *order* denotes the order of used language models. 2 + 2 means bigram word and bigram subword LM. Occupied memory after the recognition network is loaded by the decoder is in column *RAM*. 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 3 rows) compares baseline word and subword systems. The last line of the table shows hybrid system. The real time factor is estimated on Intel® Xeon® CPU, at frequency 2.33GHz.

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.

Acknowledgements. My research has been supported by Faculty of Information Technology of Brno University of Technology, by EC projects Multi-modal meeting manager (M4), No. IST-2001-34485, Augmented Multi-party Interaction (AMI), No. 506811, AMIDA (FP6-033812), by Grant Agency of Czech Republic projects No. 102/02/0124, No. 102/08/0707 and No. 102/05/0278, Czech Ministry of Interior project No. VD20072010B16, and by Czech Ministry of Defense.

References

[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] 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.

[4] 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.

[5] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiát, M. Lincoln, I. McCowan, D. Moore, V. Wan, R. Ordelman, and S. Renals. The 2005 AMI system for the transcription of speech in meetings. In *Proceedings of Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop*, Edinburgh, UK, July 2005.

[6] 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.

[7] 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.

[8] M. Mohri, F. Pereira, and M. Riley. *Speech Recognition with Weighted Finite-state Transducers*. Springer Handbook on Speech Processing and Speech Communication, Part E: Speech recognition. Springer-Verlag, Heidelberg, Germany, 2008.

[9] I. Szöke, L. Burget, J. Černocký, and M. Fapšo. Sub-word modeling of out-of-vocabulary words in spoken term detection. In *Proc. 2008 IEEE Workshop on Spoken Language Technology*, page 4. IEEE Signal Processing Society, 2008.

Selected Papers by the Author

I. Szöke, P. Schwarz, L. Burget, M. Karafiát, P. Matějka, and J. Černocký. Phoneme based acoustics keyword spotting in informal continuous speech. *Lecture Notes in Computer Science*, 2005(3658):8, 2005.

I. Szöke, P. Schwarz, L. Burget, M. Fapšo, M. Karafiát, J. Černocký, and P. Matějka. Comparison of keyword spotting approaches for informal continuous speech. In *Interspeech'2005 - Eurospeech - 9th European Conference on Speech Communication and Technology*, pages 633–636, 2005.

I. Szöke, M. Fapšo, M. Karafiát, L. Burget, F. Grézl, P. Schwarz, O. Glembek, P. Matějka, S. Kontár, and J. Černocký. But system for nist std 2006 - english. In *Proc. NIST SPoken Term Detection Evaluation workshop (STD 2006)*, page 26. National Institute of Standards and Technology, 2006.

I. Szöke, M. Fapšo, L. Burget, and J. Černocký. Hybrid word-subword decoding for spoken term detection. In *Proc. SSCS 2008: Speech search workshop at SIGIR*, page 4. Association for Computing Machinery, 2008.

I. Szöke, L. Burget, J. Černocký, and M. Fapšo. Sub-word modeling of out of vocabulary words in spoken term detection. In *Proc. 2008 IEEE Workshop on Spoken Language Technology*, page 4. IEEE Signal Processing Society, 2008.