Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 50 (2008) 980–991

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

A comparison of grapheme and phoneme-based units for Spanish spoken term detection

Javier Tejedor [a,b,*], Dong Wang [b], Joe Frankel [b], Simon King [b], José Colás [a]

[a] *Human Computer Technology Laboratory, Escuela Politécnica Superior UAM Avenue Francisco Tomás y Valiente 11, 28049, Spain*
[b] *Centre for Speech Technology Research, University of Edinburgh 2 Buccleuch Place, Edinburgh EH8 9LW, United Kingdom*

Received 31 May 2007; received in revised form 19 February 2008; accepted 17 March 2008

Abstract

The ever-increasing volume of audio data available online through the world wide web means that automatic methods for indexing and search are becoming essential. Hidden Markov model (HMM) keyword spotting and lattice search techniques are the two most common approaches used by such systems. In keyword spotting, models or templates are defined for each search term prior to accessing the speech and used to find matches. Lattice search (referred to as spoken term detection), uses a pre-indexing of speech data in terms of word or sub-word units, which can then quickly be searched for arbitrary terms without referring to the original audio.

In both cases, the search term can be modelled in terms of sub-word units, typically phonemes. For in-vocabulary words (i.e. words that appear in the pronunciation dictionary), the letter-to-sound conversion systems are accepted to work well. However, for out-of-vocabulary (OOV) search terms, letter-to-sound conversion must be used to generate a pronunciation for the search term. This is usually a hard decision (i.e. not probabilistic and with no possibility of backtracking), and errors introduced at this step are difficult to recover from. We therefore propose the direct use of graphemes (i.e., letter-based sub-word units) for acoustic modelling. This is expected to work particularly well in languages such as Spanish, where despite the letter-to-sound mapping being very regular, the correspondence is not one-to-one, and there will be benefits from avoiding hard decisions at early stages of processing.

In this article, we compare three approaches for Spanish keyword spotting or spoken term detection, and within each of these we compare acoustic modelling based on phone and grapheme units. Experiments were performed using the Spanish geographical-domain Albayzin corpus. Results achieved in the two approaches proposed for spoken term detection show us that trigrapheme units for acoustic modelling match or exceed the performance of phone-based acoustic models. In the method proposed for keyword spotting, the results achieved with each acoustic model are very similar.

Keywords: Spoken term detection; Keyword spotting; Graphemes; Spanish

1. Introduction and motivation

The increasing amount of speech and multimedia data stored electronically has motivated the development of technologies that can provide automatic search for data mining and information retrieval. These technologies have developed alongside large vocabulary continuous speech recognition (LVCSR) and use many of the same techniques.

1.1. Keyword spotting and spoken term detection

We can broadly divide audio search approaches into *keyword spotting* (KS), and lattice-based methods, which have become known as *spoken term detection* (STD). For

[*] Corresponding author. Address: Human Computer Technology Laboratory, Escuela Politécnica Superior UAM Avenue Francisco Tomás y Valiente 11, 28049, Spain.
E-mail addresses: javier.tejedor@uam.es (J. Tejedor), dwang2@inf.ed.ac.uk (D. Wang), joe@cstr.ed.ac.uk (J. Frankel), Simon.King@ed.ac.uk (S. King), jose.colas@uam.es (J. Colás).

0167-6393/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2008.03.005

keyword spotting, the terms are defined in advance, and then models or templates representing each term are used to find matches. The National Institute for Standards and Technology (NIST) introduced an STD evaluation in 2006. In doing so, they defined the task of spoken term detection as being a two-stage process in which the audio is first indexed according to word or sub-word units (e.g. phones), and then search is performed over the indexed audio. The indexing may be as a 1-best string, or as an N-best lattice.

Spanish keyword spotting systems generally use hidden Markov models (HMMs) of phone-based sub-word units (Lleida et al., 1993; Cuayahuitl and Serridge, 2002; Tejedor and Colás, 2006; Scott et al., 2007). In some cases, filler models are incorporated which represent the non-keywords in the speech (Lleida et al., 1993; Cuayahuitl and Serridge, 2002).

Lattice-based methods offer significantly faster search, as the speech is processed just once by the HMMs. A number of authors have taken the approach of searching for terms in the output of an LVCSR system (Hauptmann and Wactlar, 1997; Logan et al., 2000; Makhoul et al., 2000; Hansen et al., 2005), though a common finding is that these approaches yield high miss rates (i.e., low recall) (James and Young, 1994; Young and Brown, 1997; Tanaka et al., 2001; Yu et al., 2005). Hybrid methods based on the combination of keyword-spotting (which gives high recall) and sub-word lattice search have proven successful in combining the strengths of both methods (Yu and Seide, 2004; Tejedor and Colás, 2006).

In this paper, we present three approaches. Two of them are capable of spoken term detection, as they index the speech in terms of sub-word units. The other architecture can only perform keyword spotting, processing the audio using a recognition network composed of word models (of the keywords) and filler (garbage) models.

One of the problems which spoken-term detection must overcome is dealing with out-of-vocabulary (OOV) search terms, where we define OOV words to be those which do not appear in the pronunciation lexicon. This is important, as the OOV rate for STD in applications such as multilanguage surveillance, technical document database searching and news-story indexing tends to be higher than for transcription tasks due to a bias toward proper nouns and acronyms as search terms (Thambiratmann and Sridharan, 2007).

Search within a word-based lattice is vocabulary-dependent, as only terms which appear in the LVCSR lexicon can ever appear in the output. Therefore it is common to build lattices and employ search over sub-word units in these cases. Similar ideas have been applied to open vocabulary keyword spotting methods, for example HMM-based methods with word models composed of sub-word units. However, these methods are at the cost of considerably slower query speed, as the speech must be re-searched for each new search term (Rohlicek, 1995).

In both cases, the search term is modelled in terms of sub-word units, typically phonemes, and for OOV search terms, letter-to-sound conversion must be used to generate a pronunciation for the search term. This is usually a non-probabilistic issue and a difficult decision and errors introduced at this step are difficult to recover from. We therefore propose the direct use of graphemes (i.e., letter-based sub-word units) for acoustic modelling.

Rather than enforcing a potentially hard decision on the sequence of phone units, the relationship between graphemes and sounds will then be modelled probabilistically by the acoustic models (HMMs) themselves, rather than by an external letter-to-sound model (such as a classification tree, commonly used in text-to-speech synthesis).

This is expected to work particularly well in languages such as Spanish, where the letter-to-sound mapping is very regular. Whilst this regularity means that letter-to-sound conversion can be achieved more reliably than for some other languages (for example English), by modelling grapheme-based units directly we have the advantage of replacing a potentially error-prone hard decision with a probabilistic one which naturally accounts for this variation.

1.2. Grapheme-based automatic speech recognition (ASR)

Killer et al. (2003) demonstrated that grapheme-based LVCSR systems for Spanish can achieve performance which is close to that of phone-based systems. In some other languages – notably English, the speech sounds are harder to predict accurately from the graphemes, so grapheme-based units typically perform worse than phone-based units for acoustic modelling (Killer et al., 2003).

However, Dines and Doss (2007) show that the use of graphemes in English can yield competitive performance for small to medium vocabulary tasks in automatic speech recognition (ASR) systems. In experiments on the OGI Numbers95 task (Cole et al., 1994), a grapheme-based ASR system was found to give similar performance to the phone-based approach. However, on tasks of increased complexity, such as DARPA resource management (Price et al., 1998) and continuous telephone speech (Chen et al., 2004), the phone-based system gave lower error rates than the grapheme system.

Doss Magiami-Doss et al. (2003, 2004) also proposed the use of a phone-grapheme based system that jointly models both the phone and grapheme sub-word units during training. During decoding, recognition is performed either using one or both sub-word units. This was investigated in the framework of a hybrid hidden Markov model/artificial neural network (HMM/ANN) system. Improvements were obtained over a context-independent phone-based system using both sub-word units in recognition in two different tasks: isolated word recognition task (Magiami-Doss et al., 2003) and recognition of numbers task (Magiami-Doss et al., 2004).

1.3. Motivation and organization of this paper

Given the performance of grapheme-based models for Spanish LVCSR and the potential advantages of grapheme over phone-based units for tasks involving OOVs, we propose that grapheme-based acoustic modelling can *outperform* phone-based modelling for certain applications.

In this work, we compare grapheme-based sub-word units (monographeme and trigrapheme models) with conventional phone-based units (monophone and triphone models) for acoustic modelling using HMMs, in three different architectures for keyword spotting and spoken term detection. Sections 2 and 3 define the database and acoustic model configurations used. The three architectures are described in Section 4. Section 5 defines the evaluation metrics used, experimental results are presented in Sections 6 and 7 concludes and suggests future work.

The novel aspects of this work are in the application of grapheme-based units to acoustic modelling in keyword spotting and spoken term detection, and the confidence measures introduced in the architectures presented in Sections 4.1 and 4.3.

2. The Albayzin database

The experiments were performed using the Spanish geographical-domain Albayzin corpus (Moreno et al., 1993) which contains utterances that incorporate the names of mountains, rivers, cities, etc. Albayzin contains two separate sub-corpora: a phonetically rich component and a geographic corpus. Each of these is divided into training and test sets. We used these 4 distinct, non-overlapping portions of the data as described by Table 1.

The four sets are used as follows: The ***phonetic training set*** was used to train the acoustic models along with phone and grapheme bigram language models. The ***STD development set*** was used to train the **lexical access module** in architectures 1 and 3, and tune the language model scale and insertion penalty for the sub-word unit decoder in all three architectures. The ***phonetic test set*** was used to decide the number of Gaussian mixture components for all types of acoustic models, and the ***STD test set*** was used for final evaluation.

3. HMM-based acoustic modelling

The input signal is sampled at 16 kHz and stored with 16 bit precision. Mel-frequency cepstral coefficients were computed at 10ms intervals within 25 ms Hamming windows. Energy and first and second order derivatives were appended giving a series of 39-dimensional feature vectors.

The HTK v3.4 (Young et al., 2006) toolkit was used for the feature extraction, acoustic modelling, and decoding described in this paper.

3.1. Phone models

An inventory of 47 allophones of Spanish (Quilis, 1998) was used (as given in Appendix A), along with beginning and end of utterance silence models to build context-independent (monophone) and context-dependent (triphone) systems. All allophone and silence models had a conventional 3-state, left-to-right topology and there was an additional short pause model which had a single emitting state and a skip transition.

The output distributions for the monophone system consisted of 15-component Gaussian mixture models (GMM), and those in the triphone system used 11 components. In both cases, the number of mixture components were chosen empirically based on phone accuracy on the ***phonetic test set***. The triphone models were cross-word and were state-clustered using HTK's standard decision tree method with phonetically-motivated questions, which leads to 5632 shared states.

3.2. A grapheme inventory for Spanish

Although there is a simple relationship between spelling and sound in Spanish, care must be taken in defining the inventory of graphemes (Alarcos, 1995). We will use the term "grapheme" to mean a single unit, which is a sequence of one or more letters, to be used for acoustic modelling. This may not be precisely match the alphabet used for writing because we can expect better performance if we account for a small number of language-specific special cases.

The letter "h" only affects the phonetic realisation when it appears in the combination "ch", as in "chaqueta" ("jacket") or "Pancho" (a proper name). "ch" is always pronounced [tʃ]. Therefore "ch" is considered to be a grapheme (digrapheme in this case) and the letter "h" can be removed everywhere else. The only exceptions are in loanwords, such as "Sáhara" (borrowed from Arabic) or "hall" (borrowed from English) where the "h" is pronounced somewhere along a [h]–[χ] continuum, depending on the speaker. In the work presented here, we ignored the

Table 1
Specification of the training, development and testing sets for the Albayzin database

	Phonetic corpus (orthographically transcribed and phonetically labelled)	Geographic corpus (orthographically transcribed)
Train set	Name: ***Phonetic training set*** Contains: 4800 phonetically balanced sentences from 164 speakers: 3 h and 20 min	Name: ***STD development set*** Contains: 4400 sentences from 88 speakers: 3 h and 40 min
Test set	Name: ***Phonetic test set*** Contains: 2000 phonetically balanced sentences from 40 speakers: 1 h and 40 min	Name: ***STD test set*** Contains: 2400 sentences from 48 speakers: 2 h

pronunciation of "h" in loanwords, because the corpus used for experimentation contains no loanwords.

The combination"ll" is pronounced [ʤ] or [y], depending on context, and so is also considered a grapheme (digrapheme in this case) because its pronunciation is not related to that of its constituent letters. "ñ" is also considered a grapheme for the same reason (it is *not* an "n" plus a "~"). It is always pronounced [ɲ].

There are therefore a total of 28 grapheme units in our systems: a, b, c, ch, d, e, f, g, i, j, k, l, ll, m, n, ñ, o, p, q, r, s, t, u, v, w, x, y and z.

There are, of course, other letter combinations that could be considered as single graphemes, such as "rr", but a balance must be struck between capturing these special cases of letter-to-sound relationships, and keeping the grapheme inventory size small for statistical modelling reasons. The ideal grapheme inventory, even for a phonetically simple language like Spanish, is not easily defined. This is a challenge for grapheme-based modelling, and in future work we will consider automatic methods for avoiding a sub-optimal manual choice of grapheme inventory.

3.3. Grapheme models

The grapheme systems were built in an identical fashion to the phone-based systems; the only differences were in the inventory of sub-word units (Section 3.2) and the questions used for state clustering. The monographeme models used mixtures of Gaussians with 15 components, and the trigrapheme models used eight components. As with the phone models, these numbers were chosen empirically based on grapheme accuracy on the ***phonetic test set***. There are 3575 shared states retained after clustering in the trigrapheme system.

To build state-tied context-dependent grapheme models (trigraphemes) requires a set of questions used to construct the decision tree. There are three ways to generate those questions: using only questions about single graphemes ("singleton questions"), converting from the questions used to state-tie triphones (Section 3.1) according to a phone-to-grapheme map, or generating questions from data automatically. Killer and colleagues (Killer et al., 2003) reported that singleton questions give the best performance, so we used a singleton question set for state tying in our experiments.

4. Three architectures for keyword spotting and spoken term detection

In this work we compare three different architectures:

(1) Viterbi decoding is used to give the single most likely sequence of sub-word (phone/grapheme) units. The keyword is specified in terms of sub-word units, and a lexical access module is used to find exact or near matches in the decoded output.
(2) Viterbi decoding is used to produce an N-best sub-word (phone/grapheme) lattice. An exact word-matching procedure is applied to the lattice with the keyword specified in terms of sub-word units.
(3) Hybrid system which combines a conventional keyword spotting system composed of keywords and filler models which account for the non-keywords, with a sub-word decoder and lexical access module as in (1) in order to reduce the false alarm rate.

The three architectures (1)–(3) are described below in Sections 4.1, 4.2 and 4.3 respectively.

4.1. Architecture 1: 1-best sub-word unit decoding + lexical access

This architecture is illustrated in Fig. 1. The first processing step uses the HTK tool `HVite` to produce the single most likely (1-best) sequence of phones or graphemes, using the HMM sets trained as described in Sections 3.1 and 3.3 respectively. We refer to this as the **sub-word unit decoder**, and the output is a sequence of U phone or grapheme sub-word units $S = \{s^1, s^2, \ldots, s^U\}$.

Decoding incorporates a phone or grapheme bigram language model (LM) which was trained on the phonetic or grapheme transcription of the ***phonetic training set***, respectively Fig. 2.

These sequences are then passed to the **lexical access module** which we describe in detail in the following sections.

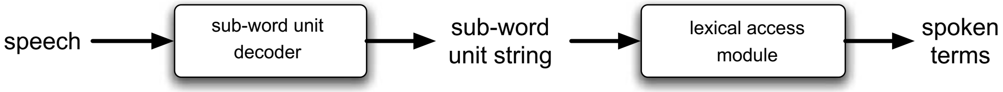


Fig. 1. The fast 1-best architecture.

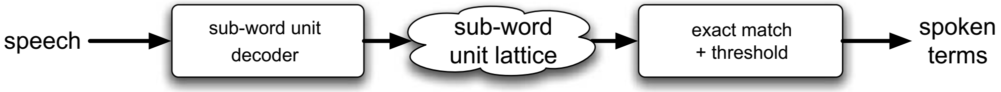


Fig. 2. The sub-word lattice architecture.

4.1.1. Lexical access module – training the alignment costs

Each keyword W is represented as a sequence of R phone or grapheme sub-word units $W = \{w^1, w^2, \ldots, w^R\}$, and search is performed within S, the output of the **sub-word unit decoder**. This approach is based on the dynamic programming algorithm proposed by Fissore et al. (1989).

The essence of the algorithm is to compute the cost of matching each keyword W with the decoded output S. The total cost is accumulated from the costs of four types of alignment error: substitution, insertion, deletion, and continuation. The first three of these are standard in ASR decoding, and 'continuation' (Fissore et al., 1989) is included in order to distinguish an insertion error from, for example, hypothesising $\{baa\}$ during a time interval in which the true phone sequence was $\{ba\}$.

Different costs are associated with each type of alignment error and are estimated as follows:

$$C_{\text{sub}}(h,k) = -\log \frac{N_{\text{sub}}(h,k)}{N_{\text{tot}}(h)} \tag{1}$$

$$C_{\text{ins}}(h,k) = -\log \frac{N_{\text{ins}}(h,k)}{N_{\text{tot}}(h)} \tag{2}$$

$$C_{\text{del}}(h) = -\log \frac{N_{\text{del}}(h)}{N_{\text{tot}}(h)} \tag{3}$$

$$C_{\text{con}}(h,k) = -\log \frac{N_{\text{con}}(h,k)}{N_{\text{tot}}(h)} \tag{4}$$

where we define:

$N_{\text{sub}}(h,k)$ is the total substitutions of test symbol k for reference symbol h; $N_{\text{ins}}(h,k)$ is the total insertions of test symbol k after reference symbol h; $N_{\text{del}}(h)$ is the total deletions of reference symbol h; $N_{\text{con}}(h,k)$ is the total continuations of test symbol k after h and $N_{\text{tot}}(h)$ is the total occurrences of reference symbol h, is given by

$$N_{\text{tot}}(h) = \sum_k [N_{\text{sub}}(h,k) + N_{\text{ins}}(h,k) + N_{\text{con}}(h,k)] + N_{\text{del}}(h) \tag{5}$$

The costs are estimated by first producing 1-best hypotheses for the training data using the **sub-word unit decoder**, and evaluating Eqs. (1)–(4) against the reference transcript. In order to develop a vocabulary-independent system, the full vocabulary of the ***STD development set*** was used. This means that many of the training keywords would be unlikely to appear in a practical application, though yields a more general system.

4.1.2. Lexical access module – finding matches

Dynamic programming is used to calculate the overall cost of matching each keyword W against the hypothesised sequence S. Letting r and u be indices for the position within W and S, respectively, the local cost function $G(r,u)$ is calculated in recursively as:

$$G(r,u) = \left| \begin{array}{l} G(r-1,u-1) + C_{\text{sub}}(w^r, s^u) \\ G(r,u-1) + C_{\text{ins/con}}(w^r, s^u) \\ G(r-1,u) + C_{\text{del}}(w^r, s^u) \end{array} \right. \tag{6}$$

where

$$C_{\text{ins/con}}(w^r, s^u) = \left| \begin{array}{ll} C_{\text{ins}}(w^r, s^u) & \text{if } s^u = s^{u-1} \\ C_{\text{con}}(w^r, s^u) & \text{otherwise} \end{array} \right. \tag{7}$$

The keyword search over a length L hypothesised sequence of sub-word units progresses as follows:

(1) For each keyword K, set the minimum window length to $W_K^{\min} = N_K/2 + 1$, where N_K is the number of sub-word units contained in the dictionary entry for keyword K. Set the maximum window length as $W_K^{\max} = W_K^{\min} + N_K$.
(2) Calculate the cost G each keyword K over each candidate window.
(3) Sort keyword hypotheses according to G, removing any for which the cost G is greater than a threshold $\Theta_{G_{\max}}$.
(4) Remove overlapping keyword hypotheses: make a pass through the sorted keyword hypotheses starting with the highest-ranked keyword, removing all hypotheses with time-overlap greater than Θ_{overlap}%.
(5) Return all keyword hypotheses with cost less than $G_{\text{best}} + \Theta_{G_{\text{beam}}}$, where G_{best} refers to the cost of the highest-ranked keyword and $\Theta_{G_{\text{beam}}}$ is beam width.

The thresholds $\Theta_{G_{\max}}$, Θ_{overlap}, and $\Theta_{G_{\text{beam}}}$, and the window sizes $W_K^{\min}$ and $W_K^{\max}$, are set on ***STD development set*** in order to give the desired trade-off of precision and recall.

As an example of the windowing in the grapheme-based approach, searching for the keyword `madrid`, which has a grapheme transcription {`m a d r i d`}, given a grapheme decoder output of {`m a i d r i e d a a n`}, the minimum and maximum windows are $W_K^{\min} = 6/2 + 1 = 4$ and $W_K^{\max} = 4 + 6 = 10$. The cost G is therefore accumulated over the following candidate windows:

{`m a i d`}, {`m a i d r`}, {`m a i d r i`}, {`m a i d r i e`}, {`m a i d r i e d`}, {`m a i d r i e d a`}, {`m a i d r i e d a a`}, {`a i d r`}, {`a i d r i`}, ..., {`i e d a`}

4.2. Architecture 2: sub-word unit lattice + exact word matching

Lattice search provides a natural extension to the 1-best path architecture above, and again search is based on sub-word (phone or grapheme) units.

The decoding process for the 1-best decoder from Section 4.1 was used, except that `HVite` was run in N-best mode. The resulting output were lattices generated from the top N tokens in each state. An example grapheme lattice is shown in Fig. 3.

4.2.1. Exact word matching in the lattice

The Viterbi algorithm provides an efficient method to find all path fragments in the lattice that exactly match the phone or grapheme string representing search terms.

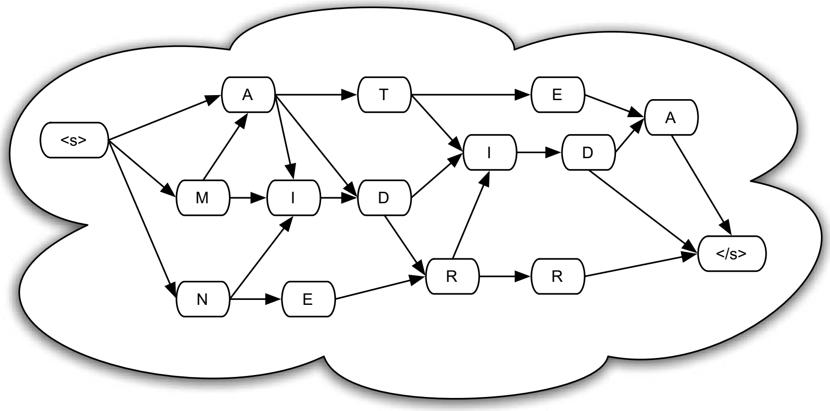


Fig. 3. Illustration of a grapheme lattice.

We use an implementation provided by collaborators at the Brno University of Technology (Szoke et al., 2005). Preliminary work confirmed the finding of Szoke et al. (2005), that given a suitably dense lattice, the accuracy improvement from allowing non-exact matches was minimal, and that $N = 5$ gave a suitably dense lattice. In the work reported here we set $N = 5$ and only consider exact matches.

For each hypothesised keyword K which the search returns, a confidence score C_K is calculated as follows:

$$C_K = L_a(K) + L(K) + L_b(K) - L_{\text{best}} \tag{8}$$

where:

- $L_a(K)$ is the log likelihood of the best path from the lattice start to the node of the first phone or grapheme of K.
- $L(K)$ is the log likelihood of keyword K, computed as the sum of the acoustic log likelihood of its constituent phones or graphemes, plus the total language model log likelihood for the sequence (weighted by the language model scale factor).
- $L_b(K)$ is the log likelihood of the best path from the last node of the last phone or grapheme of K to the end of the lattice.
- L_{best} is the likelihood of the 1-best path over the complete lattice.

$L_a(K)$ and $L_b(K)$ are computed using standard forward-backward recursions (Young et al., 2006). A threshold on the confidence score is set on the ***STD development set*** in order to reduce the false alarm rate and give the desired system performance.

4.3. Architecture 3: hybrid word + sub-word system

Standard word + filler HMM-based keyword spotting as outlined in Section 1.1 above tends to give high hit rates (recall). In the third *hybrid* architecture, we propose combining such a system with a sub-word decoder method in order to reduce the false alarms, and so increase the precision. The hybrid architecture is shown in Fig. 4.

The **sub-word unit decoder** is the same system as described above in Section 4.1: decoding takes a phone or grapheme bigram and produces the most likely sequence of phone or grapheme units.

The **keyword spotting module** uses the same set of acoustic models, though the bigram language model is replaced by a recognition network composed of words and fillers as shown in Fig. 5.

Any transition between keywords and filler models is allowed as well as self transitions for both keywords and fillers. This configuration allows multiple keywords to appear in a single utterance and multiple instances of the

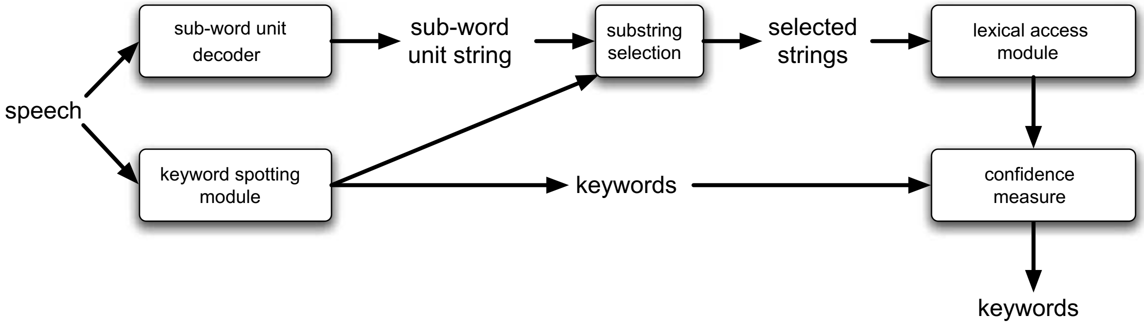


Fig. 4. The hybrid architecture.

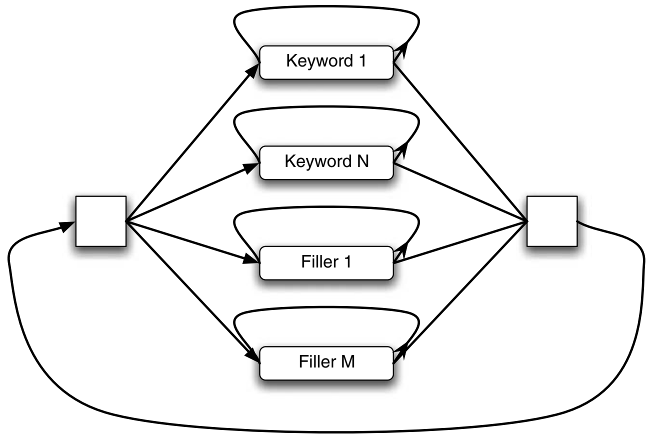


Fig. 5. The recognition network used in the **keyword spotting module** of the hybrid architecture.

same keyword in the same utterance. The keyword HMMs are constructed as concatenations of phone or grapheme HMMs, so no additional training is required. A pseudo N-gram language model, similar to the one proposed by Kim et al. (2004) was used, in which probabilities are simply assigned to the two classes of keyword and filler. The probability for the keyword class was set to be 6 and 12 times that of the fillers in the context-independent and context-dependent systems, respectively. These ratios were optimized on the ***STD development set***.

The set of hypothesised keyword matches and associated timings from the **keyword spotting module** are passed to the **substring selection module** which converts them into their corresponding phone or grapheme sequences as given by the **sub-word unit decoder**. These sub-word sequences are then processed by the **lexical access module** as described in Sections 4.1.1 and 4.1.2 above.

4.3.1. Detecting false alarms using the lexical access module

The **lexical access module** determines the cost of matching each dictionary word to the hypothesised sequence of phones or graphemes. Confidence measures can be derived from both relative and absolute cost of keywords. For example, if the second-best matching keyword has a cost which is close to that of the lowest cost keyword, then we can assign low confidence to the match. Similarly, if the absolute cost for the best matching word is high, then we also have low confidence in this match.

We adapt this idea for detection of false alarms as follows. The lexical access algorithm is run twice, first using a set of costs estimated against the keywords which were correctly detected in the ***STD development set*** by the **keyword spotting module**. This identifies a best matching word in the lexicon, along with its match cost G_{best}. In the second run of the lexical access algorithm, a set of costs trained on false alarms (FAs) produced by the **keyword spotting module** when run on the ***STD development set***, is used to return the lowest-cost word G_{FA}.

If the keywords corresponding to G_{best} and G_{FA} are the same, and $G_{\text{FA}} - G_{\text{correct}} \geqslant \alpha$, a match is returned, as we consider that the hypothesis is closer to a true keyword than a false alarm. If the words associated with G_{FA} and G_{best} differ, or the difference falls below *alpha*, the match is rejected. The threshold α is tuned on the ***STD development set***.

4.4. Vocabulary dependence

The sub-word lattice system described in Section 4.2 is the most vocabulary independent, needing no knowledge of the keywords or indeed any word list at all during training (although a dictionary is required to convert the corpus word transcription into a sub-word unit transcription). The 1-best system as described in Section 4.1 can be made independent of the keyword list, but does need to know about the corpus vocabulary during training of the **lexical access module**. The hybrid system of Section 4.3 is the most vocabulary-dependent system, and needs to know the corpus vocabulary, the keyword list and have spoken examples of the keywords during training. It is expected that the more vocabulary or corpus-dependent a system is, the better its performance should be.

5. Evaluation metrics

The purpose of this research is to identify keywords within audio. Unlike ASR, which typically considers correct recognition of all words equally important, we are interested in the trade-off of precision and recall. We use the following metrics to evaluate the systems presented in this work.

The figure of merit (FOM) was originally defined by Rohlicek et al. (1989) for the task of keyword spotting. It gives the average detection rate over the range [1, 10] false alarms per hour per keyword. The FOM values for individual keywords can be averaged in order to give an overall figure.

The NIST STD 2006 evaluation plan (NIST, 2006) defined the metrics *occurrence-weighted value* (OCC) and *actual term-weighted value* (ATWV), both of which are specifically tailored to the task of spoken term detection. These 2 metrics have been adopted and their description follows.

For a given set of terms and some speech data, let $N_{\text{correct}}(t)$, $N_{\text{FA}}(t)$ and $N_{\text{true}}(t)$ represent the number of correct, false alarm, and actual occurrences of term t respectively. In addition, we denote the number of non-target terms (which gives the number of possibilities for incorrect detection) as $N_{\text{NT}}(t)$. We also define miss and false alarm probabilities, $P_{\text{miss}}(t)$ and $P_{\text{FA}}(t)$ for each term t as:

$$P_{\text{miss}}(t) = 1 - \frac{N_{\text{correct}}(t)}{N_{\text{true}}(t)} \tag{9}$$

$$P_{\text{FA}}(t) = \frac{N_{\text{FA}}(t)}{N_{\text{NT}}(t)} \tag{10}$$

In order to tune the metrics to give a desired balance of precision versus recall, a cost C_{FA} for false alarms was defined, along with a value V for correct detections.

The occurrence-weighted value is computed by accumulating a value for each correct detection and subtracting a cost for false alarms as follows:

$$\mathrm{OCC} = \frac{\sum_{t \in \mathrm{terms}} [VN_{\mathrm{correct}}(t) - C_{\mathrm{FA}} N_{\mathrm{FA}}(t)]}{\sum_{t \in \mathrm{terms}} VN_{\mathrm{true}}(t)} \tag{11}$$

Whilst OCC gives a good indication of overall system performance, there is an inherent bias toward frequently-occurring terms.

The second NIST metric, the actual term-weighted value (ATWV) is arrived at by averaging a weighted sum of miss and false alarm probabilities, $P_{\mathrm{miss}}(t)$ and $P_{\mathrm{FA}}(t)$, over terms:

$$\mathrm{ATWV} = 1 - \frac{\sum_{t \in \mathrm{terms}} [P_{\mathrm{miss}}(t) + \beta P_{\mathrm{FA}}(t)]}{\sum_{t \in \mathrm{terms}} 1} \tag{12}$$

where $\beta = \frac{C}{V}(P_{\mathrm{prior}}(t)^{-1} - 1)$. The NIST evaluation scoring tool sets a uniform prior term probability $P_{\mathrm{prior}}(t) = 10^{-4}$, and the ratio $\frac{C}{V}$ to be 0.1 with the effect that there is an emphasis placed on recall compared to precision in the ratio 10:1.

In this work, we present results in terms of FOM and OCC. However, rather than giving the ATWV values which give point estimates of the miss and false alarm probabilities, we present these results graphically in order to show the full range of operating points. For all results, tuning for the language model scale and insertion penalty is performed on ***STD development set*** according to the metric which is used in evaluation. For all measures, higher values indicate better performance.

6. Results

The experiments were performed on the Albayzin database, described in Section 2. A set of 80 keywords were chosen based on their high frequency of occurrence and suitability as search terms for geographical-domain information retrieval, and evaluation (retrieving search terms) is performed on the ***STD test set***. Significance tests in the form of paired *t*-tests are used to compare systems, in order to determine whether differences are consistent across search terms.

6.1. Recognition accuracy

Whilst phone or grapheme recognition is not the main focus of this work, it is an important factor in STD/KS performance. We present phone accuracy results in Table 2.

Table 2
Phone and grapheme recognition accuracy for both context-independent and dependent models

	Monophone (%)	Triphone (%)	Monographeme (%)	Trigrapheme (%)
Recognition accuracy	63.9	68.2	75.2	79.1

Results are presented on the ***phonetic test set***.

For both the phone and grapheme systems, performance is improved through the use of context-dependent models. The grapheme recognition accuracy is higher, though this is expected as there are fewer graphemes than phones.

6.2. Spoken term detection and keyword spotting results

Architecture 3 uses a standard keyword spotting module in combination with a sub-word-based confidence measure. In order to examine the gain due to the confidence measure, Table 3 presents results for the keyword spotting module in isolation.

These results show that the performance improvement in moving from context-independent to context-dependent models is greater for grapheme-based models than for phones. Paired *t*-tests show that there is no systematic differences between the results of context-dependent phone and grapheme-based systems.

Table 4 presents results in terms of FOM and OCC for each of the three architectures described above in Section 4.

We first note that comparing the results of the hybrid architecture 3 with those in Table 3, the addition of the confidence measure leads to performance improvements for each metric. However, it is only for the monographeme and triphone systems evaluated under the FOM metric that the increases are statistically significant ($p < 0.01$).

6.2.1. Evaluation in terms of FOM

Table 4 shows that for evaluation in terms of FOM, context-dependent models give the best performance for all architectures and for both phone and grapheme-based models. Significance tests show that for the lattice-based approach of architecture 2, the grapheme-based systems

Table 3
Evaluation of the keyword spotting module of architecture 3 in isolation

	Keyword spotting module			
	Monophone	Triphone	Monographeme	Trigrapheme
FOM	65.9	68.3	61.0	67.6
OCC	0.74	0.73	0.66	0.78

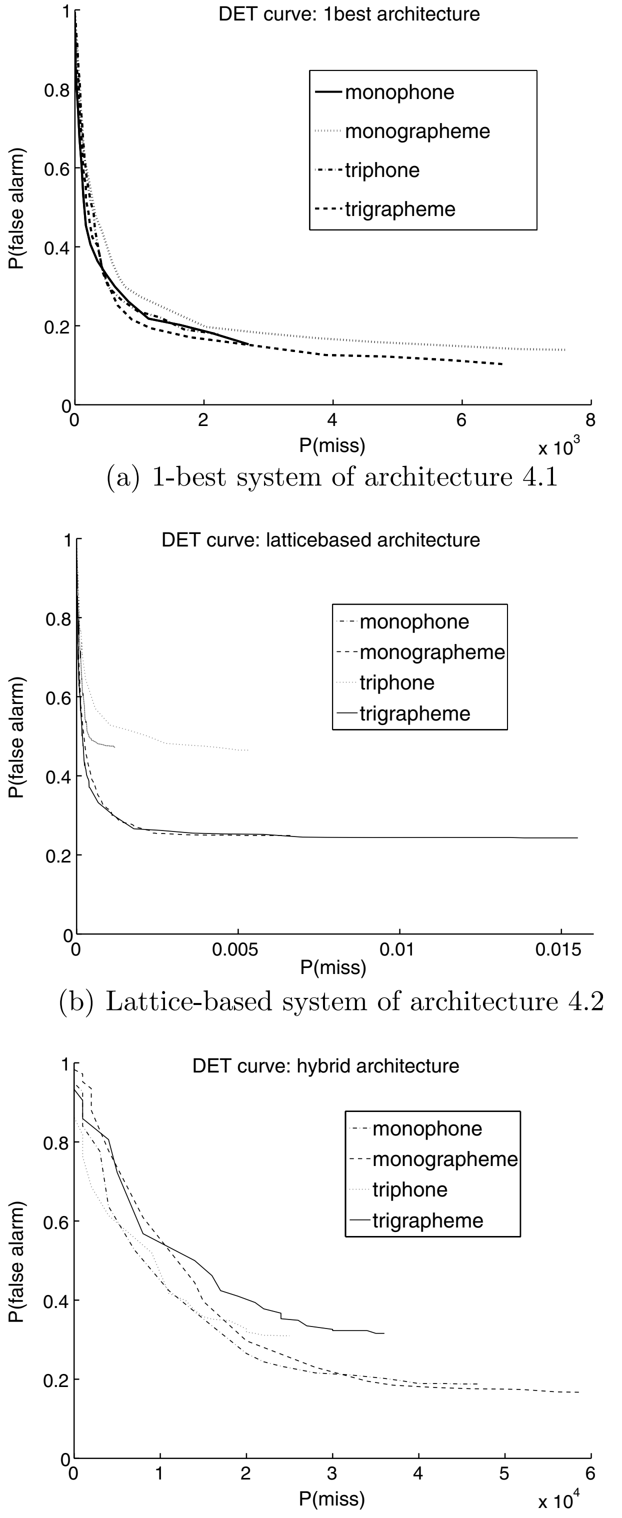
Results are given in terms of FOM and OCC for both context dependent and independent models, using grapheme and phone units.

Table 4
Results in terms of FOM and OCC for the three architectures for context-independent and -dependent phone and grapheme models

	Monophone	Triphone	Monographeme	Trigrapheme
FOM				
Architecture 1	72.7	73.5	65.9	74.4
Architecture 2	44.0	47.1	58.1	64.0
Architecture 3	80.3	82.3	76.9	79.6
OCC				
Architecture 1	0.70	0.72	0.67	0.76
Architecture 2	0.40	0.42	0.53	0.61
Architecture 3	0.85	0.84	0.84	0.85

For all measures, higher values indicate better performance.

give consistent increases in performance over the best phone-based system with $p < 0.01$. Trigraphemes gave the best performance on architecture 1, though this was not found to be statistically significant. For architecture 3, the best results are found using phone-based models, though the difference is not statistically significant.



(a) 1-best system of architecture 4.1

(b) Lattice-based system of architecture 4.2

(c) Hybrid system of architecture 4.3

Fig. 6. DET curves showing miss against false alarm probability of each architecture.

6.2.2. Evaluation in terms of OCC

We find similar patterns where the evaluation is in terms of OCC, though the performance for the phone-based models does not improve by moving from context-independent to -dependent models. Graphemes give better performance than phones for architecture 2, shown to be significant with $p < 0.01$. For architecture 3, the results are very similar, and for architecture 1, the trigrapheme gives the highest performance, though the result is not statistically significant.

6.2.3. Evaluation in terms of ATWV

We present detection error trade-off (DET) curves of the ATWV performance for each of the three architectures in Fig. 6. Each plot shows miss against false alarm probability for context-independent and dependent models, for both phone and grapheme-based systems, giving an indication of the system performance at a number of operating points.

The DET curves for architecture 1 in Fig. 6 show that the performances are quite similar for each of the systems, though the trigrapheme models marginally outperform the others for much of the range.

Fig. 6 shows the sizable performance gap between phone and grapheme-based models for the lattice-based architecture 2, and that for most of the range, the trigrapheme system provides a lower bound. It is also showed that monographeme system also outperforms both monophone and triphone systems.

DET curves for the hybrid architecture are given in Fig. 6, and show that the best performance is achieved by the monophone system.

7. Conclusions and future

Our results suggest that grapheme-based units perform at least as well as phone-based units for keyword spotting and spoken term detection, and that the relative performance of phone/grapheme models varies according to the architecture. The trends we observe when evaluating according to FOM and OCC are similar, since both are occurrence-weighted measures, whereas ATWV is term-weighted and reveals different aspects of the systems' performance. As expected, better results were found for vocabulary-dependant systems.

7.1. Hybrid approach

Architecture 3, the hybrid system, which is the most complex and the most vocabulary dependent, gives the overall best performance for each type of sub-word unit, and for each evaluation metric. The DET curves in terms of ATWV metric in Fig. 6 show that the best performance is achieved by the monophone system. At the same time the difference in FOM and OCC performance across the different acoustic models is not significant. These results may be attributed to the addition of other knowledge sources. These include the keyword network in the **keyword spotting module** and the empirically-trained costs in the **lexical access module**, which makes it more robust to weaker acoustic models. However, this architecture cannot perform spoken term detection (as defined by NIST) because it requires knowledge of the keywords when processing the speech data.

7.2. 1-Best approach

Architecture 1, the 1-best approach, is capable of spoken term detection. Again, there is not significant variation in performance across the four acoustic model types, presumably because of the additional knowledge used in the form of the **lexical access module**. However, the DET curves in terms of ATWV metric in Fig. 6 shows that the trigrapheme models marginally outperform the others for much of the range.

7.3. Lattice-based approach

Architecture 2, the lattice approach with no **lexical access module**, is the most vocabulary and corpus independent system and conforms with the requirements of recent NIST evaluations. Under this architecture we find more marked performance differences between the different acoustic models. Our experiments give evidence that for the lattice-based approach, grapheme-based systems outperform equivalent phone-based methods.

Comparing the context-independent and context-dependent systems, we find that the grapheme-based approach benefits more from context-dependent modeling than the phoneme-based approach. This is expected, as a grapheme may be pronounced quite differently according to context. By comparison, context-dependent allophones belonging to the same central phone are typically subject to a smaller degree of variation.

7.4. Grapheme-based modelling

We consider that the power of the grapheme-based system on STD tasks, especially in the lattice-based architecture, can be attributed to two factors. The first is the probabilistic description of pronunciation variation in the grapheme model, which helps represent all possible pronunciations of a search term in a single form. The second is its capacity to incorporate additional information, including both acoustic and phonological cues, in the lattice, thus improving the decision-making process in the search phase.

Grapheme-based systems do not appear advantageous under the 1-best and hybrid approaches of architectures 1 and 3, where the single most likely phone or grapheme sequences are used rather than lattices for keyword search. Given the increased acoustic variation associated with graphemes compared with phones, the advantage arises from postponing hard decisions and keeping multiple decoding paths alive. Furthermore, as stated above, the additional linguistic information from the **lexical access module** may diminish the relative performance of the different acoustic models.

7.5. Future work

Future work will include scaling up to larger tasks, which will necessitate development of language modelling techniques. One advantage of grapheme-based sub-word units is that very long span *N*-gram language models can be trained directly from very large text corpora. However, proper smoothing of these language models will be essential in order to retain the ability to model OOV search terms. Our goal is to build a full information retrieval system from the architectures presented in this work, incorporating spoken term detection and keyword spotting. Within this system, proper names will contribute to a high OOV rate, as will verbs, which present difficulties issue in Latin languages (Steigner and Schroder, 2003) such as Spanish. We will also focus in multigrapheme-based systems in order to deal with the imported graphemes when the set of keywords to search for is composed of words borrowed from other languages.

We also intend to apply architecture 2 to other languages and domains, initially English language meetings data. Whilst letter-to-sound conversion is less regular for English than Spanish, a grapheme-based approach would still be desirable given the inherent flexibility in dealing with out of vocabulary terms. The key to this will be in deriving an appropriate inventory of grapheme-based units for English, and automated methods may be required. Additionally, we are working on methods to replace the decision trees which map from CI to CD grapheme units with probabilistic mappings. This will have the effect of removing another hard decision from the system, and improve the ability to model unusual pronunciations.

Acknowledgements

This work was mostly performed whilst JT was a visiting researcher at the Centre for Speech Technology Research, University of Edinburgh, and was partly funded by the Spanish Ministry of Science and Education (TIN 2005-06885). DW is a Fellow on the Edinburgh Speech Science and Technology (EdSST) interdisciplinary Marie Curie training programme. JF is funded by Scottish Enterprise under the Edinburgh Stanford Link. SK is an EPSRC Advanced Research Fellow. Many thanks to Igor Szoke and colleagues in the Speech Processing Group of the Faculty of Information Technology at Brno University of Technology for providing the lattice search tools.

Appendix A. Phone set

a,e,i,o,u	vowel
A,E,I,O,U	stressed vowel
an,en,in,on,un	unstressed vowel between two nasals (i.e between n,m,ñ)
An,En,In,On,Un	stressed vowel between two nasals (i.e between n,m,ñ)
b	plosive at the beginning of the word or after a nasal (n,m,ñ)
B	fricative appears within a word if not after a nasal
T/	corresponds to grapheme "ch"
d	plosive at the beginning of the word or after a nasal (n,m,ñ)
D	fricative if it appears within a word, if not after a nasal
f	corresponds to grapheme "f"
g	plosive at the beginning of the sentence or after a nasal (n,m,ñ)
G	fricative if it appears within a word, if not after a nasal
X	sound as "j"
j	corresponds to grapheme "i" when "i" appears in a diphthong
J/	corresponds to grapheme "y" at the beginning of a word or after a nasal (n,m,ñ). When appearing after a word which finishes in a vowel, changes to J
J	corresponds to grapheme "y" all cases which are not considered in J/
k	corresponds to grapheme "k" and grapheme "c" when it does not sound as "z"
l	corresponds to grapheme "l"
L	corresponds to grapheme "ll"
m	corresponds to grapheme "m"
n	corresponds to grapheme "n"
N	corresponds to grapheme "n" when following a vowel.
Nn	corresponds to grapheme "ñ"
p	corresponds to grapheme "p"
r	corresponds to grapheme "r"
R	corresponds to grapheme "rr"
s	corresponds to grapheme "s"
t	corresponds to grapheme "t"
T	corresponds to graphemes "z" or "c" when sounds as "z"
w	corresponds to grapheme "u" within a diphthong
gs	corresponds to grapheme "x"

References

Alarcos, E., 1995. Gramática de la lengua española. Real Academia Española. Colección Lebrija y Bello, Espasa Calpe.

Chen, B., Cetin, O., Doddinton, G., Morgan, N., Ostendorf, M., Shinozaki, T., Zhu, Q., 2004. A CTS task for meaningful fast-turnaround experiments. In: Proceedings of Rich Transcription Fall Workshop, Palisades, New York.

Cole, R.A., Fanty, M., Noel, M., Lander, T., 1994. Telephone speech corpus development at CSLU. In: Proceedings of ICSLP, Yokohama, Japan, pp. 1815–1818.

Cuayahuitl, H., Serridge, B., 2002. Out-of-vocabulary word modeling and rejection for Spanish keyword spotting systems. In: Proceedings of MICAI, pp. 156–165.

Dines, J., Doss, M.M., 2007. A study of phoneme and grapheme based context-dependent ASR systems. In: Proceedings of MLMI, Brno, Czech Republic.

Fissore, L., Laface, P., Micca, G., Pieraccini, R., 1989. Lexical access to large vocabularies for speech recognition. IEEE Trans. Acoust. Speech, Signal Process. 37 (8), 1197–1213.

Hansen, J.R.H., Zhou, B., Seadle, M., Deller, J., Gurijala, A.R., Kurimo, M., Angkititrakul, P., 2005. Speechfind: advances in spoken document retrieval for a national gallery of the spoken word. IEEE Trans. Acoust. Speech Signal Process. 13 (5), 712–730.

Hauptmann, A., Wactlar, H., 1997. Indexing and search of multimodal information. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), Vol. 1, pp. 195–198.

James, D., Young, S., 1994. A fast lattice-based approach to vocabulary independent wordspotting. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-94), Vol. 1, pp. 465–468.

Killer, M., Stuker, S., Schultz, T., 2003. Grapheme based speech recognition. In: Proceedings of Eurospeech.

Kim, J., Jung, H., Chung, H., 2004. A keyword spotting approach based on pseudo N-gram language model. In: Proceedings of SPECOM, pp. 156–159.

Lleida, E., Marino, J., Salavedra, J., Bonafonte, A., Monte, E., Martínez, A. 1993. Out-of-vocabulary word modelling and rejection for keyword spotting. In: Proceedings of Eurospeech, pp. 1265–1268.

Logan, B., Moreno, P., Van Thong, J., Whittaker, E. 2000. An experimental study of an audio indexing system for the web. In: Proceedings of International Conference on Speech and Language Processing, Vol. 2, pp. 676–679.

Magiami-Doss, M., Stephenson, T.A., Bourlard, H., Bengio, S. 2003. Phoneme-grapheme based automatic speech recognition system. In: Proceedings of ASRU, pp. 94–98.

Magiami-Doss, M., Bengio, S., Bourlard, H., 2004. Joint decoding for phoneme-grapheme continuous speech recognition. In: Proceedings of ICASSP, Montreal, Canada, pp. 177–180.

Makhoul, J., Kubala, F., Leek, T., Liu, D., Nguyen, L., Schwartz, R., Srivastava, A., 2000. Speech and language technologies for audio indexing and retrieval. Proc. IEEE 88 (8), 1338–1353.

Moreno, A., Poch, D., Bonafonte, A., Lleida, E., Llisterri, J., Mariño, J., Nadeu, C. 1993. Albayzin speech database: design of the phonetic corpus. In: Proceedings of Eurospeech, Vol. 1, pp. 653–656.

NIST (2006). The spoken term detection (STD) 2006 evaluation plan. National Institute of Standards and Technology, Gaithersburg, MD, USA, v10 ed. <http://www.nist.gov/speech/tests/std>.

Price, P.J., Fisher, W., Bernstein, J. 1998. A database for continuous speech recognition in a 1000 word domain. In: Proceedings of ICASSP, Vol. 1, pp. 651–654.

Quilis, A., 1998. El comentario fonológico y fonético de textos. ARCO/LIBROS, S.A.

Rohlicek, J., 1995. Modern methods of Speech Processing. Kluwer, Norwell MA.

Rohlicek, J., Russell, W., Roukos, S., Gish, H., 1989. Continuous hidden Markov modeling for speaker-independent word spotting. In: Proceedings of ICASSP, Vol. 1, pp. 627–630, Glasgow, UK.

Scott, J., Wintrode, J., Lee, M., 2007. Fast unconstrained audio search in numerous human languages. In: Proceedings of IEEE International

Conference on Acoustics, Speech, and Signal Processing (ICASSP-07), Honolulu, Hawai.

Steigner, J., Schroder, M., 2003. Cross-language phonemisation in german text-to-speech synthesis. In: Proceedings of Interspeech.

Szoke, I., Schwarz, P., Matejka, P., Burget, L., Martin, K., Fapso, M., Cer- nocky, J., 2005. Comparison of keyword spotting approaches for informal continuous speech. In: Proceedings of Interspeech, Lisabon, Portugal, pp. 633–636.

Tanaka, K., Itoh, Y., Kojima, H., Fujimura, N. 2001. Speech data retrieval system constructed on a universal phonetic code domain. In: Proceedings of IEEE Automatic Speech Recognition and Understanding, pp 323–326.

Tejedor, J., Colás, J., 2006. Spanish keyword spotting system based on filler models, pseudo N-gram language model and a confidence measure. In: Proceedings of IV Jornadas de Tecnología del Habla, pp. 255–260.

Thambiratmann, K., Sridharan, S., 2007. Rapid yet accurate speech indexing using dynamic match lattice spotting. IEEE Trans. Audio Speech Process. 15 (1), 346–357.

Young, S., Brown, M., 1997. Acoustic indexing for multimedia retrieval and browsing. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), Vol. 1, pp. 199–202.

Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Amnd Povey, D., Valtchev, V., Woodland, P., 2006. The HTK Book (for HTK Version 3.4). Microsoft Corp. and Cambridge University Engineering Department.

Yu, P., Seide, F., 2004. A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech. In: Proceedings of International Conference on Speech and Language Processing, pp. 635–643.

Yu, P., Chen, K., Ma, C., Seide, F., 2005. Vocabulary independent indexing of spontaneous speech. IEEE Trans. Acoust. Speech Signal Process. 13 (5), 635–643.