Stochastic Pronunciation Modelling for Out-of-Vocabulary Spoken Term Detection

Dong Wang, *Student Member, IEEE,* Simon King, *Senior Member, IEEE,* and Joe Frankel

***Abstract*—Spoken term detection (STD) is the name given to the task of searching large amounts of audio for occurrences of spoken terms, which are typically single words or short phrases. One reason that STD is a hard task is that search terms tend to contain a disproportionate number of out-of-vocabulary (OOV) words. The most common approach to STD uses subword units. This, in conjunction with some method for predicting pronunciations of OOVs from their written form, enables the detection of OOV terms but performance is considerably worse than for in-vocabulary terms. This performance differential can be largely attributed to the special properties of OOVs.**

One such property is the high degree of uncertainty in the pronunciation of OOVs. We present a stochastic pronunciation model (SPM) which explicitly deals with this uncertainty. The key insight is to search for all possible pronunciations when detecting an OOV term, explicitly capturing the uncertainty in pronunciation. This requires a probabilistic model of pronunciation, able to estimate a distribution over all possible pronunciations. We use a joint-multigram model (JMM) for this and compare the JMM-based SPM with the conventional soft match approach. Experiments using speech from the meetings domain demonstrate that the SPM performs better than soft match in most operating regions, especially at low false alarm probabilities. Furthermore, SPM and soft match are found to be complementary: their combination provides further performance gains.

***Index Terms*—Spoken term detection, speech recognition, pronunciation modelling, letter-to-sound, out-of-vocabulary**

I. Introduction

SPOKEN term detection (STD), defined by NIST in 2006 [1], enables the searching of large quantities of audio without recourse to computationally-expensive processing of the audio signal every time a query is performed. Due to its fundamental importance in research and potential value in practice, STD has received much interest, e.g., [2]–[11].

A. Spoken term detection

The standard architecture of a STD system, as illustrated in Fig. 1, comprises an automatic speech recognition (ASR) subsystem that transcribes speech into an intermediate representation – usually word or subword lattices – and a detection subsystem that searches the lattices for query terms. In STD, a hypothesised occurrence is called a *detection*; if the detection corresponds to an actual occurrence, it is called a *hit*, otherwise it is a *false alarm (FA)*. Occurrences that are not detected by the system are called *misses*.

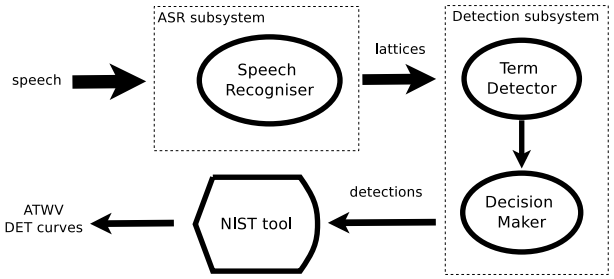

Fig. 1. The standard STD architecture: a speech recogniser converts speech into word/subword lattices; a term detector searches these lattices for potential occurrences of the search terms; a decision maker decides whether each detection is reliable. The NIST tool is used to evaluate detection performance, in terms of ATWV and DET curves.

We define a detection of a search term K as 'a finding of a partial path in the lattice that represents K', and denote it as a tuple d that encapsulates all the information available to this detection:

$$d = (K, \tau = (t_s, t_e), v_a, v_l, ...) \quad (1)$$

where v_a, v_l represent the acoustic score and language model score respectively, and τ denotes the speech segment from t_s to t_e where the detection resides. Other informative factors such as the pronunciation probability or soft match cost that we will present shortly are denoted by "...".

Each putative detection is assigned a confidence measure, or simply a *confidence*, by which the decision maker determines if the detection is reliable enough to be accepted. Letting $K_{t_s}^{t_e}$ denote the event that term K appears in the speech segment starting at time t_s and ending at time t_e, the confidence of $d = (K, \tau = (t_s, t_e), ..)$ can be evaluated by the posterior probability that the event $K_{t_s}^{t_e}$ appears given speech O. This is formulated as

$$c(d) = P(K_{t_s}^{t_e}|O) \quad (2)$$

where $c(d)$ denotes the confidence of the detection d.

In practice, $P(K_{t_s}^{t_e}|O)$ is usually computed from the lattice [12] as follows,

$$c_{lat} = \frac{\sum_{\pi_\alpha,\pi_\beta} p(O|\pi_\alpha, K_{t_s}^{t_e}, \pi_\beta) P(\pi_\alpha, K_{t_s}^{t_e}, \pi_\beta)}{\sum_\xi p(O|\xi)P(\xi)} \quad (3)$$

where π_α and π_β denote any path before and after K, with π_α starting from the beginning of the speech and π_β finishing

D. Wang was a Fellow of the EdSST Marie Curie training programme at CSTR, the University of Edinburgh. He is now a post-doc research fellow at EURECOM, France (email: Dong.Wang@ed.ac.uk). S. King is an EPSRC Advanced Research Fellow and a Reader at CSTR, University of Edinburgh (email:Simon.King@ed.ac.uk). J. Frankel was a research fellow at CSTR and now runns his own business Vegware (email: joe@cstr.ed.ac.uk).

at the end; ξ denotes any complete path through the lattice. This confidence, which we call the *lattice-based confidence* in this work, has been widely used in STD [7], [10], [13]–[15]. To distinguish it from other confidences we will introduce shortly, we denote it by c_{lat}. Overlapped detections are merged into a single detection, with the highest confidence amongst these detections being assigned to the merged detection. With confidence estimated, the decision maker asserts a detection by comparing its confidence with a threshold value that can be determined by parameter tuning on a development set.

To evaluate STD performance, NIST defines a metric called average term-weighted value (ATWV) [1], which integrates the missing and false alarm probabilities of each term into a single value and then averages over all terms. This is formulated as follows:

$$ATWV = \frac{1}{|\Delta|} \sum_{K \in \Delta} \left(\frac{N_{hit}^K}{N_{true}^K} - \beta \frac{N_{FA}^K}{T - N_{true}^K} \right) \qquad (4)$$

where Δ denotes the set of search terms and $|\Delta|$ is the number of terms in this set. N_{hit}^K and N_{FA}^K represent the number of hits and false alarms of term K respectively, and N_{true}^K is the number of actual occurrences of K in the audio. T denotes the audio length in seconds, and β is a weight factor.

Besides ATWV, NIST also uses *detection error tradeoff (DET) curves* [16] to evaluate the performance of a STD system working at various hit/FA rates. Both ATWV and DET curves are used in this paper.

Another metric commonly used in keyword spotting is the *figure of merit (FOM)*, defined as the averaged detection rate over false alarms from 0 to 10 per hour, or roughly the detection rate with 5 false alarms per hour [17]. A particular feature of FOM is that this metric tests the discriminative power of confidence measures without considering any bias, which makes it quite useful in system development, as we will discuss in Section III-B.

B. Out-of-vocabulary (OOV) term detection

Unlike conventional keyword spotting, STD is an open-vocabulary task. Queries, which are unknown at the time the system is constructed and may contain OOV words, must be handled without re-processing the speech.

OOV words are those words absent from the system dictionary. Some words are OOV simply because the system vocabulary has a fixed size, whereas others arise from the dynamics of human language. One estimate is that about 20,000 new words are coined each year [18].

In STD, *OOV terms* are those containing one or more OOV words. Terms containing only in-vocabulary words are called *in-vocabulary (INV) terms*. OOV terms present a significant challenge to STD; in one real spoken document retrieval system, 12% of queries contained OOV terms [19]. Since new words are continually being created, even a very large, but fixed, vocabulary STD system will eventually receive a significant numbers of OOV queries.

C. Motivations

The usual approach to detecting OOV terms employs subword units [6], [8], [20]: search terms are converted to a subword sequence (usually phonemes) by letter-to-sound (LTS) conversion. This sequence is then searched for in previously-generated subword lattices or transcripts [11], [20]. In this paper, we use a phoneme-based system.

STD performance is usually much worse for OOV terms than INV terms. Reasons for this include more speech transcription errors and incorrect pronunciation predictions. We hypothesise that OOV terms have a particularly high degree of uncertainty in pronunciation, more phonetic/phonotactic diversity, and are more weakly modelled by the acoustic and language models.

We hypothesise that OOV STD can be improved by addressing these special properties of OOV terms. Here, we focus on the high degree of pronunciation uncertainty. Different from INV detection, this uncertainty to a large extent comes from less standardised pronunciations used by speakers (e.g., because they are unfamiliar with the OOV words). We call this *lexical deviation*, which is quite different from acoustic variation and therefore can not be fully compensated for by commonly employed soft match techniques (e.g. [21]–[24]).

This paper presents a stochastic pronunciation modelling (SPM) approach to deal with lexical deviation. In this approach, we use a probabilistic pronunciation model to predict all possible pronunciations of a search terms if it is OOV, and then search for all these pronunciations in term detection; this amounts to treating pronunciation as a hidden variable, and integrating it out. The confidence of a detection is then composed from the confidence given by the pronunciation model and the usual confidence from lattice search. We implement the SPM using a joint-multigram model.

Compared to our previous work [25], [26], we are now able to present a clearer understanding of the particular variation exhibited in pronunciations of OOV terms, including a subjective experiment to illustrate this; we also now propose a complete theory of stochastic pronunciation modelling and report more reliable experimental results than previously given.

In the rest of the paper, we start by discussing the issue of pronunciation uncertainty, focusing on lexical deviation of OOV terms. In Section III, we present the SPM and show how to use it to deal with lexical deviation; we also compare SPM with soft match and show that these two techniques are complementary and can be combined. An implementation of SPM based on a joint-multigram model (JMM) is presented in Section IV. In Section V, we describe our experiments and report results. Section VI concludes with some thoughts on future work.

II. OOV PRONUNCIATION UNCERTAINTY

A. OOV Uncertainty

Uncertainty is ubiquitous in speech and is a major challenge to STD, particularly with respect to OOV term detection. OOV terms (i.e., their phonemic pronunciations) are more likely to be misrecognised and pronunciation prediction usually suffers

from a high error rate [27]–[31]. Speakers, when encountering a novel term, vary their speaking style: they may slow down, examine the spelling structure, guess the pronunciation, hesitate, and so on. This leads to more acoustic variation. The pronunciations chosen for OOV terms may vary between speakers more than for INV terms, leading to lexical deviation which does not exist in INV terms. The interaction between acoustic variation and lexical deviation makes OOV terms rather difficult to deal with.

The lattice-based approach [32]–[35] is widely used to mitigate recognition errors. Better LTS models can be used, such as joint-multigram models [6]. Soft match is the most common technique for mitigating acoustic variation; it allows for some mismatch between the pronunciation predicted for the search term and the phoneme sequences in the lattice and typically involves a penalty based on either edit distance [13], [36], [37], acoustic confusion [21], [22], [24], [38] or model distance [39], [40]. Lexical deviation, however, has not been widely investigated until recently [3], [25].

B. Acoustic variation and lexical deviation

Compared to acoustic variation that has been widely recognised for some time [41]–[45], lexical deviation is less of concern. For illustration, consider the word 'Buccleuch', a Scottish place name and part of the street name where our research group once resided. It is a typical OOV term and its correct pronunciation is not always correctly predicted by speakers unfamiliar with the word. We surveyed 100 participants via Amazon's Mechanical Turk (mTurk). For each word in a list of 50 OOV terms including, we asked them to select a single pronunciation from a list we provided; pronunciations were represented using the IPA (which was explained to them by way of example words). The results for the word 'Buccleuch' are shown in Table I, where 'Pron.' denotes the pronunciation variants, and '#' denotes the number of participants that chose each variant. We see that there is not a single predominant pronunciation; on the contrary, people selected a variety of pronunciations, leading to lexical deviation.

Although both are 'pronunciation variation' and are interconnected, lexical deviation differ from acoustic variation in several ways. Firstly, acoustic variation arises during speech production, while lexical deviation arises during speech planning. Secondly, acoustic variation is subtle and is affected by factors such as environment, emotion, speaking rate, etc., whereas lexical deviation is perhaps more stable within a given speaker but varies across speakers, for reasons including demographic factors such as native language, social status, etc. Finally, acoustic variation can be compensated for by soft match, but lexical deviation can only be properly compensated for in the pronunciation prediction model.

TABLE I
VARIOUS PRONUNCIATIONS OF WORD 'BUCCLEUCH'

Pron.	#	Pron.	#	Pron.	#
buklju:	15	buklu:	36	bʌklu:	9
bʌklju:	7	buklju:ʧ	14	bʌklju:ʧ	10
bʌklu:ʧ	3	bʌklju:ʧ	3	others	3

Lexical deviation can be described by a probabilistic distribution over pronunciations that we might expect from speakers when uttering OOV words or terms in the data being searched. Fig. 2 summarises these pronunciation distributions of the 50 OOV terms we surveyed through mTurk. It can be seen that many terms have several pronunciations and that representing each with only the single most likely pronunciation would fail to account for a substantial probability mass. Further analysis shows that names of foreign cities and technical terms tend to be more confusing and thus demonstrate more variability in pronunciation. This motivates the stochastic pronunciation modelling approach that we present in the next section.

III. STOCHASTIC PRONUNCIATION MODELLING

A. Stochastic pronunciation modelling

Motivated by the finding that each pronunciation variant of an OOV term is spoken by a certain proportion of speakers, we propose to use the probability distribution $P(Q|K)$ to model lexical deviation. $P(Q|K)$ is the probability that term K is pronounced using pronunciation Q, and so can be called a *stochastic pronunciation model*.

We now use the pronunciation model $P(Q|K)$ to deal with lexical deviation in OOV term detection. Motivated by the idea that a detection based on *any possible pronunciation* of a search term might contribute a correct detection, we consider all possible pronunciations during lattice search. In order to represent detections that are found as the result of different pronunciations, we first extend the definition of a detection to be

$$d = (K, Q, \tau, v_a, v_l, ...) \tag{5}$$

where Q is the pronunciation which lead to detection d. With this extended definition, we immediately notice that the lattice-based confidence of (3) should be defined as the posterior

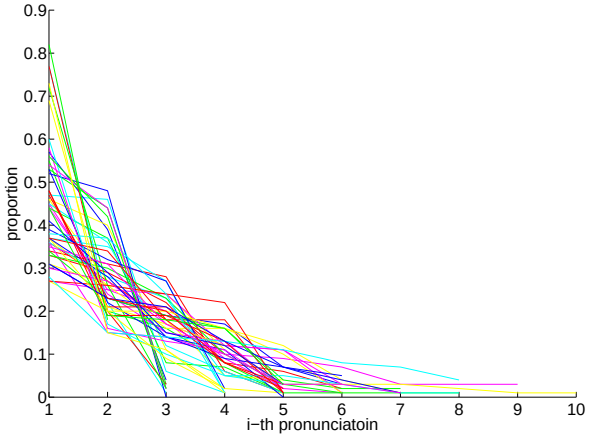

Fig. 2. Pronunciation distribution of 50 OOV terms, surveyed from 100 participants through mTurk. The x-axis represents the different pronunciations of each term in descending order of usage frequency and the y-axis represents the relative usage frequency.

probability of pronunciation Q instead of term K, given the speech:

$$c_{lat}(d) = P(Q_{t_s}^{t_e}|O) \tag{6}$$

where $Q_{t_s}^{t_e}$ denotes pronunciation Q appearing in the speech segment from time t_s to time t_e, where t_s and t_e are obtained from the tuple d.

Furthermore, different pronunciations have different prior probabilities, which must be taken into account by forming a *composite confidence*.

The composite confidence can be derived from a hierarchical speech generation framework, in which a term K randomly generates a pronunciation Q following $P(Q|K)$, and pronunciation Q randomly generates a speech segment O following $p(O|Q)$. The event posterior probability $P(K_{t_s}^{t_e}|O)$ in (2) then can be factorised as follows:

$$\begin{aligned} P(K_{t_s}^{t_e}|O) &= \sum_Q P(K_{t_s}^{t_e}, Q|O) && (7)\\ &= \sum_Q P(K, Q_{t_s}^{t_e}|O) && (8)\\ &= \sum_Q P(Q_{t_s}^{t_e}|O)P(K|O,Q) && (9)\\ &= \sum_Q P(Q_{t_s}^{t_e}|O)P(K|Q) && (10) \end{aligned}$$

where Q represents any possible pronunciation of K. A 'layer separation' assumption has been applied in deriving (10) from (9): we assume K and O are independent given Q. (10) indicates that detecting a search term equals to detecting all its pronunciations, and the confidence of a term occurring within a speech segment can be obtained by summing the confidences of the detections of all its pronunciations, if we define the confidence of a detection as follows:

$$c(d) = P(Q_{t_s}^{t_e}|O)P(K|Q). \tag{11}$$

where K and Q are obtained from the tuple d.

In practice, we find modelling $P(Q|K)$ ('letter-to-sound') gives better performance than model $P(K|Q)$ ('sound-to-letter')[1]. Assuming that $P(Q)$ and $P(K)$ have uniform distributions, we arrive at:

$$c(d) = P(Q_{t_s}^{t_e}|O)P(Q|K). \tag{12}$$

Note that $P(Q_{t_s}^{t_e}|O)$ is just the lattice-based confidence defined in (6), and $P(Q|K)$ represents lexical deviation. We will call $P(Q|K)$ the *pronunciation confidence*:

$$c_{pron}(d) = P(Q|K) \tag{13}$$

where Q and K are obtained from the tuple d. Now the composite confidence of detection d of term K found with pronunciation Q can be written as:

$$c_{spm}(d) = c_{lat}(d)^{1-\gamma} c_{pron}(d)^{\gamma} \tag{14}$$

[1]This might be because these models are trained on dictionaries, which are always organised as a mapping from spelling to pronunciations.

where we have introduced an interpolation factor γ to balance the contribution of c_{lat} and c_{pron}. Note that the pronunciation confidence represents lexical deviation and is given by a stochastic pronunciation model. Therefore, we call this approach stochastic pronunciation modelling (SPM).

According to (10), all detections of K that share the same starting and ending time should be merged as a single detection with their confidences being summarised. In practice, the possibility that two pronunciations of a term are detected in the same speech segment is not significant, so we simply assign the highest confidence to the merged detection, i.e.,

$$P(K_{t_s}^{t_e}|O) = \max_i c_{spm}(d_i) \tag{15}$$

where $d_i = (K, Q_i, \tau = (t_s, t_e), ...)$. This approximation simplifies the term search algorithm and is consistent with our approach to dealing with overlapped detections presented in Section I-A.

B. SPM and confidence bias

The composite confidence derived in the previous section does not necessarily lead to optimal STD. The decision maker (Fig. 1) determines whether a detection is a reliable hit or a false alarm: this is a binary classification task with ATWV (defined in (4)) as the loss function. According to decision theory, an optimal decision for this task requires an *unbiased* classification posterior probability $P(C_{hit}|d)$ where C_{hit} denotes the hit class. Any other confidence biased from $P(C_{hit}|d)$ is invalid, even if it possesses the same discriminative power as $P(C_{hit}|d)$.

If we assume the speech transcription (i.e., the subword lattice) includes all possible non-K terms (i.e., all terms that lead to d as a false alarm), then the event posterior probability $P(K_{t_s}^{t_e}|O)$ can be regarded as $P(C_{hit}|d)$. In practice, however, this is not always true: both the system dictionary and language model are limited, which means that $P(K_{t_s}^{t_e}|O)$ is very likely to be biased with respect to $P(C_{hit}|d)$. Moreover, any additional assumptions and approximations will introduce further biases. For example, we assumed that $P(Q)$ and $P(K)$ are uniform in order to derive (11), the lattice-based approach itself is an approximation, and the maximisation in (15) is another approximation. These assumptions and approximations cause the composite confidence to be biased with respect to the ideal classification posterior probability; this may lead to suboptimal STD performance.

We have shown in previous work [46] that a linear remedy can be used to ameliorate the bias problem for the lattice-based confidence. With SPM, however, the bias problem is more significant. On one hand, more assumptions have been introduced; on the other hand, the interpolation factor inevitably changes the value of the confidence (see (14)). The highly biased confidence makes optimising the linear remedy rather challenging, especially when this optimisation is interweaved with the optimisation of the interpolation factor.

To solve this problem, we designed a two-step optimisation approach: first the interpolation factor γ is selected to optimise the discriminative power of the composite confidence, then the parameters of the linear remedy are selected to optimise

the loss function, ATWV. Since the linear remedy does not change the discriminative power of the confidence, we arrive at a confidence that is optimal in the sense of both discriminative power and bias. Since the FOM metric concerns discriminative power only, we use it as the objective function of the first optimisation.

C. Soft match

A widely used approach to pronunciation uncertainty treatment is soft match. This approach allows a degree of mismatch between the phoneme sequence for search (i.e., pronunciation) and the detected phoneme sequence, so that it is able to compensate for pronunciation variations and transcription errors. To conduct a comparative study for SPM, we implemented the soft match approach as well.

In order to allow soft match, we extend the definition of a detection as follows,

$$d = (K, Q, \hat{Q}, \tau, v_a, v_l, ...) \tag{16}$$

where Q is the pronunciation and $\hat{Q}$ is the detected phoneme sequence. The lattice-based confidence, therefore, should be re-defined as follows,

$$c_{lat}(d) = P(\hat{Q}_{t_s}^{t_e}|O) \tag{17}$$

where $\hat{Q}_{t_s}^{t_e}$ denotes that the detected phoneme sequence $\hat{Q}$ starts from time t_s in the audio and ends at time t_e.

As in the SPM-based approach, we derive the composite confidence of a detection with soft match from the hierarchical speech generation framework, in which a term K generates a *predicted* pronunciation Q, and Q randomly generates a *found* phoneme sequence $\hat{Q}$, following $P(\hat{Q}|Q)$; finally $\hat{Q}$ randomly generates a speech segment O following $P(O|\hat{Q})$. Starting from the event posterior probability $P(K_{t_s}^{t_e}|O)$, we have,

$$\begin{aligned} P(K_{t_s}^{t_e}|O) &= \sum_{\hat{Q}} P(\hat{Q}_{t_s}^{t_e}, Q|O) && (18)\\ &= \sum_{\hat{Q}} P(\hat{Q}_{t_s}^{t_e}|O)P(Q|O,\hat{Q}) && (19)\\ &= \sum_{\hat{Q}} P(\hat{Q}_{t_s}^{t_e}|O)P(Q|\hat{Q}) && (20)\\ & && (21) \end{aligned}$$

where Q has replaced K as it is determinately generated by K, and a layer separation assumption has again been applied to get from (19) to (20). Similarly to SPM, this indicates that detecting a term (or its pronunciation) is equal to detecting all the pronunciations allowed by soft match, and the confidence of detecting the term can be computed by summing the confidences of all detections being found with these 'soft matched' pronunciations sharing the same starting and ending time, if we define the confidence of a single detection as:

$$c(d) = P(\hat{Q}_{t_s}^{t_e}|O)P(Q|\hat{Q}) \tag{22}$$

where Q and $\hat{Q}$ are obtained from the tuple d. Note that $P(\hat{Q}_{t_s}^{t_s}|O)$ is the lattice-based confidence, and $P(Q|\hat{Q})$ represents the match degree between Q and $\hat{Q}$.

Again, let us assume uniform $P(Q)$ and $P(\hat{Q})$ and introduce an interpolation factor μ; the composite confidence is then obtained as follows,

$$\begin{aligned} c_{soft}(d) &= P(\hat{Q}_{t_s}^{t_e}|O)^{1-\mu}P(\hat{Q}|Q)^{\mu} && (23)\\ &= c_{lat}(d)^{1-\mu}c_{match}(d)^{\mu} && (24) \end{aligned}$$

where $c_{soft}(d)$ indicates that it is a composite confidence based on soft match, and

$$c_{match}(d) = P(\hat{Q}|Q) \tag{25}$$

has been explicitly defined to represent the degree of match between the *found* and *predicted* pronunciations, which we call the *match confidence*.

A widely used approach to compute $P(\hat{Q}|Q)$ is based on a confusion matrix [21]–[24], [47]–[50]. In this approach, the insertion/deletion/substitution probabilities of phoneme pairs are estimated by a forced alignment between phoneme recognition output on the development set and the canonical transcription, which forms a confusion matrix that represents the match degree of a phoneme pair (a special null phoneme is included to allow insertions and deletions). $P(\hat{Q}|Q)$ is then computed as accumulation of the match degrees of the phoneme pairs of Q and $\hat{Q}$, which is obtained from a forced alignment.

Just like SPM, soft match suffers a bias problem in confidence estimation; therefore the two-step optimisation approach should be applied here as well.

D. SPM and soft match combination

Comparing SPM and soft match, we note that they deal with pronunciation uncertainty differently: SPM operates on the lexical level (acoustic information is not considered; the pronunciation model is trained on the lexicon and captures common patterns across different words) while soft match deals with acoustic variation (lexical information is not considered; the confusion matrix is trained on transcribed speech). Therefore, we might expect to obtain further improvements by combining them. A simple way to do this is to apply soft match when conducting SPM-based detection, and integrate the pronunciation confidence and match confidence with the lattice-based confidence as follows,

$$c(d) = c_{lat}^{1-\gamma-\mu}c_{pron}^{\gamma}c_{match}^{\mu}. \tag{26}$$

This *integration approach* tends to cause many false alarms, because such a wide range of pronunciation variety is allowed. In addition, the composite confidence tends to be more biased than that of either SPM or soft match alone, which makes it unlikely to be fully compensated by a linear remedy. A possible solution is to constrain the variety and allow just one sort of variation: either that caused by SPM or that caused by soft match. This equates to conducting SPM and soft match detection individually, then merging the detections found by the two systems: overlapped detections are merged as a single detection, and the highest confidence is assigned to the merged detection. We call this the *combination approach*.

IV. JOINT-MULTIGRAM MODEL-BASED SPM

The stochastic pronunciation model must estimate the pronunciation probability distribution $P(Q|K)$. Multiple pronunciation dictionaries [42], [43] could be used, but would of course only be able to estimate $P(Q|K)$ for in-vocabulary words. In this paper, we propose to use a joint-multigram model (JMM) as the stochastic pronunciation model, since it can estimate $P(Q|K)$ for any word or term.

A. JMM-based 1-best pronunciation prediction

The joint-multigram model, proposed by [51], has been demonstrated to be superior to other models for LTS, e.g., [31], [52]. Motivated by the idea that writing and speaking are independently derived from an underlying hidden process of human language, a joint-multigram model represents a probability distribution over sequences of phoneme-grapheme joint units.

Following the notation of Bisani and Ney [53], we call a grapheme-phoneme joint unit a *graphone*, denoted by $u = (\tilde{g}, \tilde{q})$ where $\tilde{g}$ and $\tilde{q}$ are the grapheme and phoneme component of u respectively. Both $\tilde{g}$ and $\tilde{q}$ contain a sequence of symbols whose length is from N_{min} to N_{max}. With graphones defined, the joint probability of spelling G and pronunciation Q can be written in graphones U as:

$$P(G,Q) = \sum_{U;G(U)=G,Q(U)=Q} P(U) \quad (27)$$

$$= \sum_{U;G(U)=G,Q(U)=Q} P(u_1, u_2, ..., u_K) \quad (28)$$

where U is the concatenation of u_1, u_2,...,u_K, and $G(U)$ and $Q(U)$ denote the grapheme and phoneme component of U, respectively. The task of pronunciation prediction is then formulated as follows:

$$\hat{Q}(G) = \arg\max_{Q} P(G,Q) \quad (29)$$

$$= \arg\max_{Q} \sum_{U;G(U)=G,Q(U)=Q} P(U). \quad (30)$$

Similar as [31], [52], we factor $p(U)$ into graphone n-grams:

$$P(U) = \prod_{j=1}^{|U|} P(u_j|h_j) \quad (31)$$

where $|U|$ is the length of the graphone sequence U, h_j is the graphone history of u_j.

To improve the prediction accuracy, we extend the basic algorithm in two ways: *insertion compensation* to compensate for long pronunciations; *backward decoding* to make use of right-context dependence [54].

We trained and tested the JMM on the dictionary used by the AMI RT05s LVCSR system [55], with 36575 words randomly selected out for training, 4064 words for parameter tuning and 8000 words for evaluation. Various graphone sizes (N_{min} and N_{max}) and n-gram models were examined, and various smoothing techniques for the n-gram model were explored. The experimental results show that the best performance is obtained when setting $N_{min} = 1$ and $N_{max} = 2$ when applying a 4-gram graphone model smoothed by Kneser-Ney discounting and interpolation. Details of the experiments and results can be found in [54].

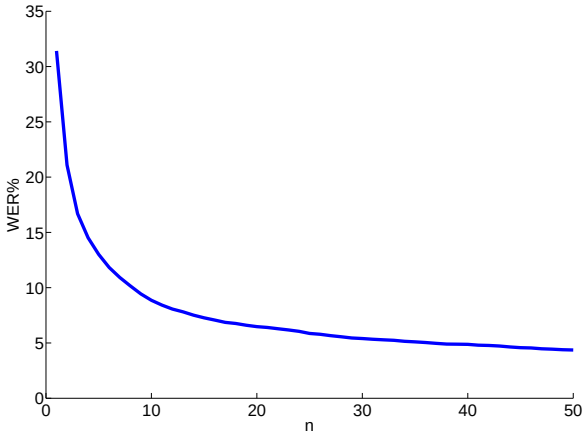

Fig. 3. The results of JMM-based n-best pronunciation prediction in terms of n-best WER. An n-best error means none of the n-best predictions is correct.

The experimental results of 1-best pronunciation prediction are shown in Table II, in terms of word error rate (WER). For comparison, performance using a class and regression tree (CART) model – the default method for LTS used by the Festival speech synthesis system [56] – is reported as well. JMM generally outperforms the CART.

B. JMM-based n-best pronunciation prediction

Now we employ the JMM to predict n-best pronunciations. The method is to keep n paths in each step when searching for pronunciations, and then chose the n pronunciations with highest confidence when the search is complete.

The confidence of a pronunciation Q is defined as the probability $P(Q|G)$, which can be derived from the posterior probabilities of the paths that correspond to Q in the decoding lattice constructed in the search process:

$$P(Q|G) = \frac{\sum_{G(U)=G,Q(U)=Q} P(U)}{\sum_{U' \subseteq \Re(K)} P(U')} \quad (32)$$

where $\Re(K)$ stands for the decoding lattice for term K and $P(U)$ denotes the probability of graphone path U in $\Re(K)$.

The results of the n-best pronunciation prediction are shown in Fig. 3. From this figure, we can see that correct pronunciation of most terms is found within the top few candidates in the n-best list.

TABLE II
RESULTS OF 1-BEST PRONUNCIATION PREDICTION

Model	WER (%)
CART	35.2
joint multigram	33.2
+ insertion compensation	32.7
+ reverse decoding	31.3

C. JMM-based stochastic pronunciation model

We now use the JMM to predict pronunciations for spoken term detection, by defining

$$P(Q|K) = P(Q|G_K) \tag{33}$$

where G_K denotes the spelling of the search term K.

SPM requires all possible pronunciations to be considered during term search (because we are effectively integrating out a hidden variable), whereas the JMM is only able to provide an n-best list. Considering that memory and computation requirements increase with n, and looking at the results in Fig. 3, we chose n=50 and assumed that this is equivalent in practice to considering all possible pronunciations.

V. EXPERIMENTS

A. Experimental settings

We selected the meeting domain in which to conduct our experiments because there are realistic applications for STD in this domain (e.g., search and indexing involving novel terms), large amounts of speech are available and ASR in this domain is challenging.

To ensure the OOV terms in the experiment have similar properties to genuine novel terms that could be expected in a real application, we defined OOV terms strictly as: those containing no words listed in the dictionaries of the ASR system or of the term detector, and not appearing in the training material for either the acoustic or language models. To create a list of OOV terms, we compared the AMI dictionary (recently created, in active use and so assumed to represent current usage) and the COMLEX Syntax dictionary v3.1 (published by LDC in 1996 and therefore historical from a STD perspective). We selected 412 terms from the AMI dictionary that do not occur in the COMLEX dictionary. We also added another 70 *artificial* OOV terms (which occur more frequently) that are plausible search terms. This results in 482 search terms having a total of 2736 occurrences in the evaluation data. These terms were removed from the system dictionaries; furthermore, all utterances and sentences that contain these terms were deleted from the speech and text training corpora. This ensures that they were entirely unseen during system training and tuning.

The speech data used in this work for acoustic model (AM) training, system development and performance evaluation are from multi-participant meetings recorded in several institutes, including the International Computer Science Institute (ICSI), the National Institute for Standards and Technology (NIST), the Carnegie Mellon University Interactive Systems Laboratory (ISL), the Linguistic Data Consortium (LDC), the Virginia Polytechnic Institute and State University (VT) and partners of the AMI project. The speech recorded using individual head-mounted microphones (known as the IHM condition) was used. After OOV purging, 122744 utterances (80.2 hours) of speech was available to train the AM. The RT04s development set was used for parameter tuning. The evaluation set comprised the RT04s and RT05s eval sets and a new meeting corpus recorded recently at the University of Edinburgh in the AMIDA project, totalling 11 hours of speech.

TABLE III
STD PERFORMANCE USING CART VS. 1-BEST JMM

Model for LTS	ATWV	max-ATWV
CART	0.2126	0.2607
1-best JMM	**0.2761**	0.2770

The text corpus used to train the language model (LM) was kindly provided by the AMI project and is the same as used by the AMI RT05s large vocabulary continuous speech recognition (LVCSR) system [55]. It contains text from various sources such as news and transcripts of speech corpora, plus a large amount of text collected from the web, totally 521.4 million words after OOV purging. A 50k word dictionary from the AMI project (also OOV purged) was used to convert the word-based text corpus to a phoneme-based one. The same dictionary was also used to train the joint-multigram model following the procedure discussed previously. This JMM was then used as the stochastic pronunciation model to predict pronunciations for OOV terms.

We built a phoneme-based STD system. The ASR subsystem was built using the speech and text data described above. The acoustic models were 3-state triphone HMMs employing conventional 39-dim MFCC features, with cepstral mean and variance normalisation (CMN + CVN) applied. A 6-gram phoneme LM was used to perform speech decoding (this LM order was selected empirically). The averaged density of the resulting lattices is 805 nodes per second.

The HTK toolkit was used to train the acoustic models and transcribe speech to lattices and the SRI LM toolkit was used to train graphone and phoneme n-gram models. The term detector was implemented with *Lattice2Multigram* generously provided to us by the Speech Processing Group, FIT, Brno University of Technology. Term-dependent normalisation [46] was applied in all experiments. The metrics used to evaluate STD performance are ATWV and DET curves; ATWV values with the optimal balance of P_{miss} and P_{FA} are presented as well, denoted by *max-ATWV*.

B. STD using 1-best prediction from a joint multigram model

We first examine STD performance with 1-best pronunciations predicted by the joint-multigram model. The term search is based on exact match, i.e., no mismatch is allowed. For comparison, the same experiment is conducted with the CART model implemented in Festival. The ATWV results are shown in Table III, which show that the JMM-based pronunciation prediction clearly outperforms the CART-based prediction in OOV STD. A pairwise t-test shows the this improvement is statistically significant ($p < 0.001$). The JMM 1-best system based on exact match is the baseline in the following experiments.

C. STD with SPM

In this section we test the SPM approach. As a special case, we first examine the performance with n-best pronunciations predicted by the JMM. Various values of n are examined, and for each n, a pruning threshold η on prediction confidence is

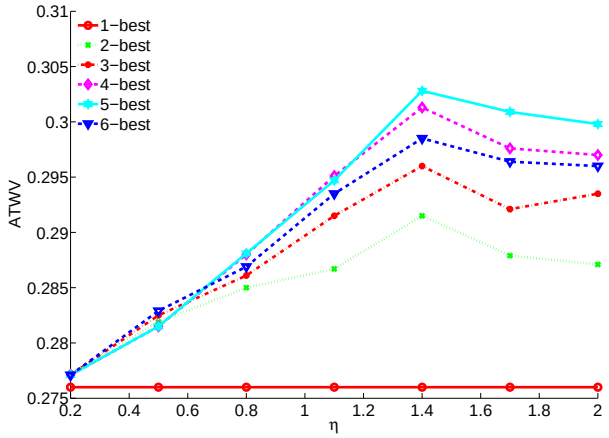

Fig. 4. STD performance using n-best pronunciation predictions from the JMM. Each curve represents the ATWV results for a particular value of n, as the confidence pruning threshold η is varied.

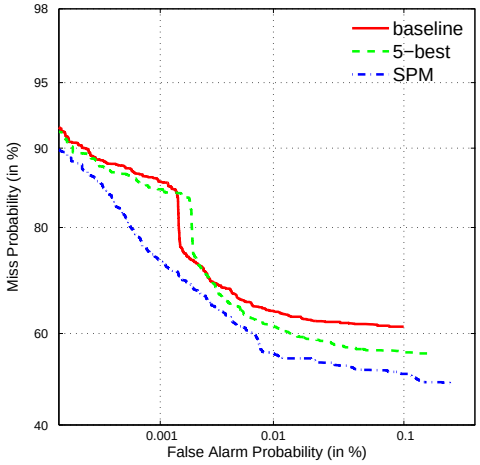

Fig. 5. DET curves for 1-best vs. 5-best vs. SPM.

applied to remove unlikely pronunciations. The ATWV results are shown in Fig. 4, from which we can see that the 5-best prediction with $\eta = 1.4$ gives the best performance. A t-test shows that all the n-best systems (for $n > 1$) significantly outperform the 1-best system ($p < 0.01$).

Now we extend the n-best approach to a full SPM treatment. In theory, SPM considers all possible pronunciations in term search; in practice, however, resources are limited and performance gains with too many pronunciations become marginal, so we just consider the best 50 pronunciations in experiments, assuming that this is a sufficient approximation to the full distribution over all pronunciations. This assumption is supported by the results in Fig. 3, in which we can see clearly that little additional improvement is obtained by considering pronunciations more than 50.

The two-step optimisation approach discussed in III-B is applied to optimise the interpolation factors and the linear remedy with the development set, which shows that $\gamma = 0.98$ is the optimal setting.

The results are shown in Table IV. We can see that in terms of ATWV, the SPM-based system substantially outperforms the baseline system which is based on 1-best prediction, and the 5-best system which is based on 5-best prediction. A t-test shows the SPM-based system performs significantly better when compared with both the 1-best system ($p < 10^{-5}$) and the 5-best system ($p < 10^{-4}$).

TABLE IV
STD RESULTS FOR 1-BEST VS. 5-BEST VS. SPM

System	ATWV	max-ATWV	FOM
baseline	0.2761	0.2770	38.89
5-best	0.3028	0.3040	41.30
SPM	**0.3415**	0.3586	46.87

Fig. 5 shows the DET curves of the 1-best, 5-best and the SPM-based systems. We can see that the SPM-based system systematically outperforms the other two systems in the entire operation area. A particular interesting point is that even with a low FA probability, SPM still works well, although it is reasonable to hypothesise that more false alarms might be caused. This result indicates that SPM is a 'safe' approach to addressing lexical diversity: it finds more term occurrences but does not reduce detection accuracy.

D. STD with soft match

For soft match, we compensate for both insertions, deletions and substitutions, and found that compensating only for substitutions provided the best performance improvement. This might be attributed to the fact that allowing all kinds of mismatches produce too many false alarms, and the heterogenous match confidence produces bias that is more difficult to remedy. For that reason, we just allow substitutions in our soft match approach, and control the maximum number of substitutions allowed.

The two-step optimisation approach is applied to optimise the interpolation factor μ and the linear remedy, which shows that $\mu = 0.99$ is the optimal setting in spite of the maximum number of substitutions allowed.

The results are shown in Table V where the maximum number of substitutions allowed is shown in brackets. We can see that soft match generally improves STD performance substantially over the baseline system (which is based on exact match). A t-test shows that the improvement with soft mach is always statistically significant ($p < 0.01$ with any of the three soft match -based systems). Comparing the three soft match systems, we find that the FOM and max-ATWV values can be increased by allowing more substitutions, indicating that the ideal performance of the system has been improved. However it is the system that allows maximum one substitution reports the best ATWV, which suggests that in practice, a suitable threshold is more difficult to find by parameter tuning if the pattern of mismatch becomes complex.

The DET curves of the soft-match based systems are shown in Fig. 6; for comparison, the baseline system (1-best prediction, exact match) and SPM-based systems (50-best prediction, exact match) are also presented. We find that soft match performs the best when the FA probability is high; however,

TABLE V
STD RESULTS FOR EXACT MATCH VS. SOFT MATCH

System	ATWV	max-ATWV	FOM
baseline	0.2761	0.2770	38.89
soft match(1)	**0.3468**	0.3617	47.25
soft match(2)	0.3415	0.3812	49.73
soft match(3)	0.3421	0.3827	50.35

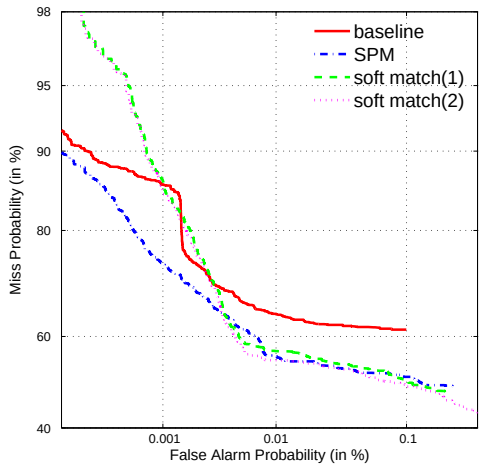

Fig. 6. DET curves for baseline vs. SPM vs. soft match.

TABLE VI
STD RESULTS FOR SPM AND SOFT MATCH COMBINATION

System	Approach	ATWV	max-ATWV
SPM+soft match(1)	integration	0.0555	0.1727
SPM+soft match(1)	combination	0.3762	0.3795
SPM+soft match(2)	combination	**0.3768**	0.3918
SPM+soft match(3)	combination	0.3758	0.3849

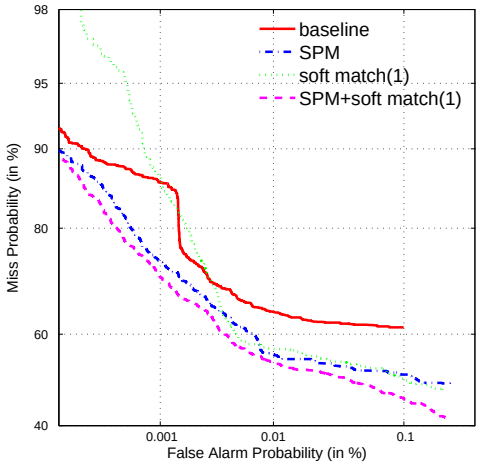

Fig. 7. DET curves for baseline vs. SPM vs. soft match vs. SPM+soft match.

in the area of a low FA probability, soft match performs much worse than SPM, even than the baseline. This is somewhat expected, as soft match allows pronunciation variants with little constraint so that pronunciations may be considered even if they are totally impossible. SPM, on the contrary, is constrained by pronunciation rules (represented by the JMM model) and hence only considers those 'legal' pronunciations. This explains why SPM shows a good performance at various hit/FA rates, while soft match works only when the FA probability is high.

E. SPM and soft match combination

SPM and soft match deal with pronunciation uncertainty in different ways and display different behaviours (seen in the DET curves), which suggests system combination. As presented in Section III-C, either an integration approach or combination approach can be used to combine these two techniques. The ATWV results are shown in Table VI. We can see that the integration approach performs rather poor, for which the reason we have discussed already. With the combination approach, which is a constrained version of the integration approach, significant performance improvement is attained ($p < 0.01$). This applies to the combination with all the three soft match -based systems.

Fig. 7 shows the DET curves of the systems based on SPM and soft match, and their combination. We see clearly that the combination system performs better than each individual system in all the operation area. This suggests the combination approach is also a safe technique to enhance OOV STD. For simplicity, we just present the soft match -based system allows maximum one substitution in Fig. 7, but this conclusion applies to all the three soft match -based systems.

VI. CONCLUSION

We have presented a stochastic pronunciation modelling approach for STD which is able to deal with the pronunciation uncertainty of OOV terms. Compared to the conventional soft match approach, which only compensates for acoustic pronunciation variation, the SPM approach can handle lexical deviation which arises from inconsistent pronunciations of OOV terms. We experimented with an SPM approach based on a joint-multigram model and compared it with the soft match approach. Experimental results show that the SPM is superior to soft match when the FA probability is low; this is the most interesting region of operation for many applications. Furthermore, we demonstrated that the two techniques are complementary and their combination gives additional performance gain.

One future work is to refine the pronunciation model. Although the joint-multigram model performs well, the true distribution of pronunciations for OOVs is undoubtedly complex, because it arises from the behaviour of speakers when they guess the pronunciation of less familiar words. We are exploring a new pronunciation model based on a condition random field (CRF); preliminary results are encouraging.

ACKNOWLEDGMENT

This work was carried out while DW was a Fellow on the EdSST interdisciplinary Marie Curie training programme at CSTR, University of Edinburgh. This work used the Edinburgh

Compute and Data Facility which is partially supported by eDIKT. Special thanks to the AMI and AMIDA projects for releasing their data and sharing resources. The revision has also been supported by the French Ministry of Industry (Innovative Web call) under contract 09.2.93.0966, "Collaborative Annotation for Video Accessibility" (ACAV).

REFERENCES

[1] NIST, *The spoken term detection (STD) 2006 evaluation plan*, 10th ed., National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, September 2006. [Online]. Available: http://www.nist.gov/speech/tests/std

[2] J. Mamou and B. Ramabhadran, "Phonetic query expansion for spoken document retrieval," in *Proc. Interspeech'08*, Brisbane, Australia, September 2008, pp. 2106–2109.

[3] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraclar, "Effect of pronunciations on OOV queries in spoken term detection," in *Proc. ICASSP'09*, Taipei, Taiwan, April 2009, pp. 3957–3960.

[4] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. Workshop on Searching Spontaneous Conversational Speech (SIGIR-SSCS'07)*, Amsterdam, July 2007.

[5] D. Vergyri, A. Stolcke, R. R. Gadde, and W. Wang, "The SRI 2006 spoken term detection system," in *Proc. NIST spoken term detection workshop (STD 2006)*, Gaithersburg, Maryland, USA, December 2006.

[6] M. Akbacak, D. Vergyri, and A. Stolcke, "Open-vocabulary spoken term detection using graphone-based hybrid recognition systems," in *Proc. ICASSP'08*, Las Vegas, Nevada, USA, March 2008, pp. 5240–5243.

[7] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system," in *Proc. Interspeech'07*, Antwerp, Belgium, August 2007, pp. 2393–2396.

[8] I. Szöke, M. Fapšo, M. Karafiát, L. Burget, F. Grézl, P. Schwarz, O. Glembek, P. Matějka, J. Kopecký, and J. Černocký, "Spoken term detection system based on combination of LVCSR and phonetic search," in *Machine Learning for Multimodal Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 4892/2008, pp. 237–247.

[9] I. Szöke, L. Burget, J. Černocký, and M. Fapšo, "Sub-word modeling of out of vocabulary words in spoken term detection," in *Proc. IEEE Workshop on Spoken Language Technology (SLT'08)*, Goa, India, December 2008, pp. 273–276.

[10] S. Meng, P. Yu, J. Liu, , and F. Seide, "Fusing multiple systems into a compact lattice index for Chinese spoken term detection," in *Proc. ICASSP'08*, Las Vegas, Nevada, USA, March 2008, pp. 4345–4348.

[11] K. Thambiratmann and S. Sridharan, "Rapid yet accurate speech indexing using dynamic match lattice spotting," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 346–357, January 2007.

[12] F. Wessel, K. Macherey, and R. Schlüter, "Using word probabilities as confidence measures," in *Proc. ICASSP'98*, vol. 1, Seattle, Washington, USA, May 1998, pp. 225–228.

[13] D. R. H. Miller, M. Kleber, C. lin Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech'07*, Antwerp, Belgium, August 2007, pp. 314–317.

[14] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. ACM-SIGIR'07*, Amsterdam, The Netherlands, July 2007, pp. 615–622.

[15] 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'06)*. Gaithersburg, Maryland, USA: National Institute of Standards and Technology, December 2006.

[16] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech'97*, vol. 4, Rhodes, Greece, September 1997, pp. 1895–1898,.

[17] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proc. ICASSP'89*, Glasgow, UK, May 1989, pp. 627–630.

[18] D. Watson, *Death Sentence, The Decay of Public Language*. Knopf, Sydney, 2003.

[19] B. Logan, P. Moreno, J.-M. V. Thong, and E. Whittaker, "An experimental study of an audio indexing system for the web," in *Proc. ICSLP'00*, vol. 2, Beijing, China, October 2000, pp. 676–679.

[20] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. ACM-SIGIR'07*, 2007, pp. 615–622.

[21] K. Ng, "Subword-based approaches for spoken document retrieval," Ph.D. dissertation, MIT, February 2000.

[22] K. Audhkhasi and A. Verma, "Keyword search using modified minimum edit distance measure," in *Proc. ICASSP'07*, vol. 4, Honolulu, Hawaii, USA, April 2007, pp. 929–932.

[23] J. Pinto, I. Szöke, S. Prasanna, and H. Heřmanský, "Fast approximate spoken term detection from sequence of phonemes," in *Proc. The 31st Annual International ACM SIGIR Conference*. Singapore: Association for Computing Machinery, July 2008, pp. 28–33.

[24] R. Wallace, R. Vogt, and S. Sridharan, "spoken term detection using fast phonetic decoding," in *Proc. ICASSP'09*, Taipei, Taiwan, April 2009, pp. 4881–4884.

[25] D. Wang, S. King, and J. Frankel, "Stochastic pronunciation modelling for spoken term detection," in *Proc. Interspeech'09*, Brighton, UK, September 2009, pp. 2135–2138.

[26] D. Wang, S. King, J. Frankel, and P. Bell, "Stochastic pronunciation modelling and soft match for out-of-vocabulary spoken term detection," in *Proc. ICASSP'10*, Texas, US, March 2010.

[27] R. Damper and J. Eastmond, "Pronunciation by analogy: Impact of implementational choices on performance," *Language and Speech*, vol. 40, no. 1, pp. 1–23, 1997.

[28] A. W. Black, K. Lenzo, and V. Pagel, "Issues in building general letter to sound rules," in *Proc. 3rd ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia, 1998, pp. 77–80.

[29] W. Daelemans, A. van den Bosch, and J. Zavrel, "Forgetting exceptions is harmful in language learning," *Machine Learning*, vol. 34, no. 1-3, pp. 11–41, 1999.

[30] P. Taylor, "Hidden Markov models for grapheme to phoneme conversion," in *Proc. Interspeech'05*, Lisbon, Portugal, September 2005, pp. 1973–1976.

[31] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[32] D. A. James and S. J. Young, "A fast lattice-based approach to vocabulary independent wordspotting," in *Proc. ICASSP'94*, Yokohama, Japan, September 1994, pp. 377–380.

[33] M. Brown, J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. Young, "Open-vocabulary speech indexing for voice and video mail retrieval," in *Proc. ACM Multimedia conference*, Boston, MA, 1996.

[34] S. J. Young, M. Brown, J. T. Foote, G. J. F. Jones, and K. Spärck Jones, "Acoustic indexing for multimedia retrieval and browsing," in *Proc. ICASSP'97*, vol. 1, Munich, Bavaria, Germany, April 1997, pp. 199–202.

[35] F. Seide, P. Yu, C. Ma, , and E. Chang, "Vocabulary-independent search in spontaneous speech," in *Proc. ICASSP'04*, vol. 1, Montreal, Quebec, Canada, May 2004, pp. 253–256.

[36] K. Thambiratnam and S. Sridharan, "Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting," in *Proc. ICASSP'05*, vol. 1, Philadelphia, Pennsylvania, USA, March 2005, pp. 465–468.

[37] J. Mamou, Y. Mass, B. Ramabhadran, and B. Sznajder, "Combination of multiple speech transcription methods for vocabulary independent search," in *Proc. Workshop on Search in Spontaneous Conversational Speech (SIGIR-SSCS'08)*, Singapore, 2008.

[38] U. Chaudhari, H.-K. J. Kuo, and B. Kingsbury, "Discriminative graph training for ultra-fast low-footprint speech indexing," in *Proc. Interspeech 2008*, Las Vegas, Nevada, USA, March 2008, pp. 2175–2178.

[39] Y. Itoh, T. Otake, K. Iwata, K. Kojima, M. Ishigame, K. Tanaka, and S. wook Lee, "Two-stage vocabulary-free spoken document retrieval-subword identification and re-recognition of the identified sections," in *Proc. ICSLP'06*, Pittsburgh, USA, September 2006, pp. 1161–1164.

[40] K. Iwata, K. Shinoda, and S. Furui, "Robust spoken term detection using combination of phone-based and word-based recognition," in *Proc. Interspeech'08*, Brisbane, Australia, September 2008, pp. 2195–2198.

[41] T. Sloboda and A. Waibel, "Dictionary learning for spontaneous speech recognition," in *Proc. ICSLP'96*, Philadelphia, USA, October 1996, pp. 2328–2331.

[42] N. Cremelie and J.-P. Martens, "In search of better pronunciation models for speech recognition," *Speech Communication*, vol. 29, no. 2-4, pp. 115–136, 1999.

[43] H. Strik and C. Cucchiarini, "Modeling pronunciation variation for ASR: A survey of the literature," *Speech Communication*, vol. 29, no. 5, pp. 225–246, 1999.
[44] Y. R. Oh, J. S. Yoon, and H. K. Kim, "Acoustic model adaptation based on pronunciation variability analysis for non-native speech recognition," *Speech Communication*, vol. 49, no. 1, pp. 59–70, 2007.
[45] T. Hain, "Implicit pronunciation modelling in ASR," in *Proc. ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexical Adaptation for Spoken Language*, Estes Park, CO, 2002.
[46] D. Wang, S. King, J. Frankel, and P. Bell, "Term-dependent confidence for out-of-vocabulary term detection," in *Proc. Interspeech'09*, Brighton, UK, September 2009, pp. 2139–2142.
[47] K. Ng, "Towards robust methods for spoken document retrieval," in *Proc. ICSLP'98*, Sydney, Australia, November 1998, pp. 939–942.
[48] M. Wechsler, E. Munteanu, and P. Schäuble, "New techniques for open-vocabulary spoken document retrieval," in *Proc. ACM SIGIR 1998*, Melbourne, Australia, August 1998, pp. 20–27.
[49] S. Srinivasan and D. Petkovic, "Phonetic confusion matrix based spoken document retrieval," in *Proc. The 23rd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'00)*, New York, NY, USA, 2000, pp. 81–87.
[50] A. Amir, A. Efrat, and S. Srinivasan, "Advances in phonetic word spotting," in *Proc. The 10th International conference on information and knowledge management (CIKM'01)*, Atlanta, Georgia, USA, November 2001, pp. 580–582.
[51] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Proc. Eurospeech'95*, Madrid, Spain, September 1995, pp. 2243–2246.
[52] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proc. Eurospeech'03*, Geneva, Switzerland, September 2003, pp. 2033–2036.
[53] M. Bisani and H. Ney, "Investigations on joint-multigram models for grapheme-to-phoneme conversion," in *Proc. ICSLP'02*, Denver, USA, September 2002, pp. 105–108.
[54] D. Wang, "Out-of-vocabulary spoken term detection," Ph.D. dissertation, The Center for Speech Technology Research, Edinburgh University, December 2009.
[55] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, J. Vepa, and V. Wan, "The AMI meeting transcription system: Progress and performance," in *Machine Learning for Multimodal Interaction*. Springer Berlin/Heidelberg, 2006, vol. 4299/2006, pp. 419–431.
[56] R. A. J. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

Dong Wang received the B.Sc. and M.Sc. in computer science at Tsinghua Univ. in 1999 and 2002, and then worked for Oracle China in 2002-2004 and IBM China in 2004-2006. He joined CSTR, University of Edinburgh in 2006 as a research fellow and PhD student supported by a Marie Curie fellowship, from where he received his Ph.D. in 2010. His is now working in EURECOM France as a post-doc fellow.

Simon King (M'95, SM'08) received M.A.(Cantab) and M.Phil. degrees in Engineering from the University of Cambridge in 1992 and 1993 and a Ph.D. from the University of Edinburgh in 1998. He is a Reader in Linguistics and English Language and his interests include speech synthesis, recognition and signal processing. He serves on ISCA SynSIG committee, co-organises Blizzard Challenge, was recently an assoc. ed. of IEEE Trans. Audio, Speech & Lang. Proc., is on the IEEE SLTC and the editorial board of Computer Speech and Language.

Joe Frankel (M'05) graduated with first class honours in Mathematics and Statistics from Edinburgh University in 1998. A background in probabilistic modelling paved the way for a PhD place at Centre for Speech Technology Research (CSTR). By the time he had completed his PhD in summer 2003, he had gained a strong interest in the application of machine learning techniques to automatic speech recognition. He now runs his own business Vegware (http://www.vegware.com).