Context Dependent Acoustic Keyword Spotting Using Deep Neural Network

Guangsen WANG and Khe Chai SIM
*School of Computing, National University of Singapore, Singapore
E-mail: {guangsen,simkc}@comp.nus.edu.sg

***Abstract*—Language model is an essential component of a speech recogniser. It provides the additional linguistic information to constrain the search space and guide the decoding. In this paper, language model is incorporated in the keyword spotting system to provide the contexts for the keyword models under the weighted finite state transducer framework. A context independent deep neural network is trained as the acoustic model. Three keyword contexts are investigated: the phone to keyword context, fixed length word context and the arbitrary length word context. To provide these contexts, a hybrid language model with both word and phone tokens is trained using only the word n-gram count. Three different spotting graphs are studied depending on the involved contexts: the keyword loop graph, the word fillers graph and the word loop fillers graph. These graphs are referred to as the context dependent (CD) keyword spotting graphs. The CD keyword spotting systems are evaluated on the Broadcasting News Hub4-97 F0 evaluation set. Experimental results reveal that the incorporation of the language model information provides performance gain over the baseline context independent graph without any contexts for all the three CD graphs. The best system using the arbitrary length word context has the comparable performance to the full decoding but triples the spotting speed. In addition, error analysis demonstrates that the language model information is essential to reduce both the insertion and deletion errors.**

I. Introduction

Speech has been used as the primary approach of information exchange and social communications for human beings since prehistory. In addition to human-human interaction, spoken language communication is adopted in human-machine interaction as well. Compared to the conventional way of human-computer interaction, e.g., keyboard strokes, mouse clicks, speech is a much more straightforward way. With the ubiquitous mobile devices, speech manifests its another appealing advantage of offering the hand-free communication scheme between the users and the devices. One of the well-known such applications is the "voice search" by both Microsoft Bing and Google search. They can already meet the requirement of recognising and understanding simple speech search queries. For these voice search engines, it is more important to reliably spot the keywords than to get the full transcriptions of the spoken utterance.

The task of key word spotting (KWS) is to search for various query words or terms in a large collection of heterogeneous audio archives rapidly and accurately. KWS thus provides a satisfactory audio mining solution for spoken document retrieval tasks. Therefore, it is widely used in the on-line applications like real-time stream monitoring, as well as offline tasks like data mining and indexing.

Most of the current keyword spotting systems uses the sub-word models, for example, the phone or syllables to represent the keyword models and the fillers models based on the Hidden Markov Models (HMM). The detection of the keywords depends solely on the acoustic models. In other words, no context of the keywords is modelled. On the other hand, for the ASR system, language model is a standard component to constrain the Viterbi search and provides the context information for the decoder. For the keyword spotting tasks, the introduction of the language model information also has two potential benefits: it can provide a better model for the non-keyword speech and a statistical context for the keywords. Incorporating simple language model information to weigh the non-keyword models has shown to enjoy significant improvement [1], [2], [3] . In [2], the language model was used to provide the transition probabilities for the sub-grammars. Similarly, in [3], a bigram syllable/keyword language model was built using the training transcriptions to serve as the transition probabilities between the keywords and the syllable fillers. In this paper, we aim to investigate the context dependent keyword spotting systems with the context information provided by the language model. Three different language model contexts are investigated including the phone to keyword context, the fix-length word context, and the arbitrary length word context. The context configurations used in this paper are given below:

- Context independent: This configuration is the commonest used in the current keyword spotting literature. No context of the keywords are modelled. The keyword spotting system depends solely on the acoustic models. The graph is shown in Figure 2. The fillers/background models are the monophone loops.
- Phone to keyword context: A straight-forward approach of incorporating context to the context independent graph is to weigh the transitions between the background monophones and the keywords, as well as the keyword to keyword transitions using the language model scores. This is very similar to the work in [3] where syllables are used as fillers and syllable to keyword transitions are weighted using a hybrid syllable/keyword language model.
- Fixed length word to word context: Inspired by the usage of the language model in the ASR system, we propose to

add higher word level contexts to the keyword spotting network in Figure 3. Instead of using the monophone fillers or the syllable fillers, two word fillers are used. The transitions between the left/right fillers and the keyword models are weighted by the word language model. The context expansion length shown in Figure 3 is fixed as three. In other words, only the immediate left and right word are used as the contexts of the middle keyword. The additional contexts given by the language model can guide the search to help the detection of the keyword.

- Arbitrary length word to word contexts: As a generalisation of the fixed length word context, the left and right context length can be with arbitrary lengths as shown in Figure 4. This is realised by using two word loops as the left and right contexts of the middle keyword.

The acoustic model used in this paper is a context independent deep neural network (DNN) [4], [5]. The training targets are obtained from the forced alignment with monophone states. The weighted finite state transducer (WFST) [6] is used to implement all the spotting networks and decoding. A hybrid language model with both word and phone tokens is trained using only the word n-gram count. Through the WFST composition, the hybrid language model can be incorporated into various context dependent keyword spotting networks to provide all the necessary contexts.

The remaining of the paper is organised as follows: a literature review is firstly given in section 2. The overview of a keyword spotting system is given in section 3. The components of a typical keyword spotting network and how they can be represented as WFSTs are given in section 4. The incorporation of the context information is presented in section 5. The training of the hybrid language model is detailed in section 6. The keyword spotting performance is evaluated in section 7 followed by a detailed error analysis in section 8. Section 9 summarises the findings and concludes the paper.

II. Related Work

The earliest work on KWS mainly uses template-based dynamic time-warping (DTW) techniques [7], [8]. The major limitation for these approaches is that the spotting is forced to adhere to some local time duration constraints of the keyword templates. With the development of Hidden Markov Models (HMM) for speech recognition, they are also widely adopted for the keyword spotting task. Depending on how the keyword models are modelled, the HMM based keyword spotting systems can be roughly categorised as whole-word based, sub-word based and large vocabulary based. Whole word based approach is one of the earliest HMM based model for keyword spotting [9], [10]. The keywords are modelled as HMMs trained from the utterances of the keywords. The garbage model is also modelled as an HMM trained with non-keyword speech data. The training of the keyword model assumes there is enough training data containing the keywords, which is not always available. Consequently, sub-word based word spotters are introduced [1], [11]. In these systems, the keywords are modelled as a concatenation of corresponding sub-words, e.g., phones, syllables. The garbage model is modelled as a loop of all the sub-word models. This approach does not require the training data to contain the keywords thus solving the main issue of the word based systems. However, since both the keyword and garbage models use the same sub-word units, the garbage model has the potential of modelling all the words including the keywords. Hence, the tuning of insertion penalties is often needed although this can be somehow circumvented by using carefully designed garbage models [11]. The large vocabulary continuous speech recognition (LVCSR) based approaches rely on some additional linguistic constraints to improve the spotting performance [12]. The keyword spotting is performed on either the lattices [13], [14] or the transcriptions [2] generated by a LVCSR system. The limitation of this approach is that its computational cost implied by the large vocabulary decoding.

In addition to the HMM based systems, hybrid neural network (NN) and Hidden Markov Models [15], [16] are used to estimate the state posteriors of the HMMs for the keyword spotting tasks. Compared to the HMM based approach, NNs do not make any independent assumptions on the statistical distributions. They accommodate discriminative training naturally. They also tend to offer a much more compacted model. More recently, recurrent neural networks (RNNs) are used in the keyword spotting tasks. Long short-term memory (LSTM) RNNs or Bidirectional LSTM (BLSTM) networks are shown to be a promising technique for improving the keyword spotting performance by modelling temporal contexts [17].

High-level linguistic information through recognition grammars can provide some contexts for the keyword and constrain the search space. Incorporating such prior knowledge has been shown to improve the spotting performance significantly compared to the non-grammar constrained approaches. However, they are mostly used for fixed or well-defined queries [18], [2], where these queries can be easily described by a set of word sequences and represented using some finite state grammars. In [18], a finite state grammar (FSG) is constrained to a set of most frequently appeared query patterns in an auto-attendant system. In the similar vein, "sub-grammars" combined with a non-keyword model are used in [2] to describe the queries for their event spotting system.

III. Context Dependent Keyword Spotting Overview

The keyword is usually known beforehand for the keyword spotting task. Therefore, KWS can be seen as as a special case of speech recognition with a vocabulary size of two, namely, the keyword(s) and the non-keyword. The general framework of a typical keyword spotting system is shown in Figure 1. For unconstrained KWS, the input sequence is assumed as an unconstrained sequence of background and extraneous speech modelled by the left filler followed by the keywords and then followed by another unconstrained sequence of background and extraneous speech modelled as the right filler. In addition, a background model is also usually adopted to model all the non-keyword speeches to increase the spotting robustness.

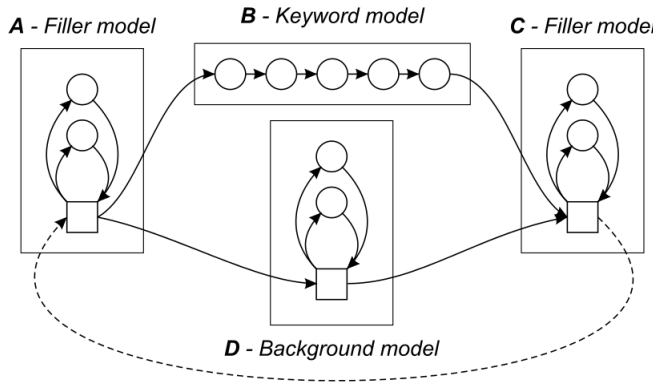

Fig. 1. General framework of keyword spotting.

TABLE I
Illustration of KWS with different contexts

Word sequence	Of POLITICAL SCIENCE
Correct phone sequence	/ah/ /v/ /p/ /ah/ /l/ /ih/ /t/ /ah/ /k/ /ah/ /l/ /s/ /ay/ /ah/ /n/ /s/
Context independent	/ah/ /v/ /p/ /l/ /ah/ /t/ /k/ /ah/ /l/ /s/ /ay/ /n/ /s/
Phone to keyword context	/ah/ POLITICAL /s/ /ay/ /n/ /s/
Fix length word to word context	/ah/ POLITICAL SCIENCE
Arbitrary length word to word context	OF POLITICAL SCIENCE

In our paper, the following formula is used in our WFST based keyword spotting system:

$$\mathcal{L}_{left} + \mathcal{L}(KW) + \mathcal{L}_{right} - \mathcal{L}_{bkg} \geq \beta \tag{1}$$

where $\mathcal{L}$ is the log likelihood obtained from the acoustic model, β is a threshold. Moving the background model score to the RHS, we have:

$$\mathcal{L}_{left} + \mathcal{L}(KW) + \mathcal{L}_{right} \geq \beta + \mathcal{L}_{bkg} \tag{2}$$

The LHS corresponds to the upper path of Figure 1 and the RHS corresponds to the lower path. Therefore, the keyword spotting task becomes simply choosing the best path in the decoding network. Since negative log weights are used in the WFST framework, the threshold β can be well represented as the insertion penalties imposed on the background model transitions as shown in the lower path of Figure 1.

Context information from the language models can be incorporated into the network to aid the keyword spotting as discussed in the introduction. Useful contexts include phone to word context and word to word context. Table III shows an example of how the context information can help the keyword spotting. The table shows the keyword spotting results of the keyword "POLITICAL" in the phrase "OF POLITICAL SCIENCE" with different context information provided by the language model. The context independent KWS system depends solely on the acoustic scores during Viterbi search. The phone sequence of "/p/ /l/ /ah/ /t/ /k/ /ah/ /l/" has the highest acoustic score and is the output of the context independent keyword spotting system. The correct pronunciation "/p/ /ah/ /l/ /ih/ /t/ /ah/ /k/ /ah/ /l/" does not have the highest acoustic score. Therefore, the corresponding keyword "POLITICAL" is missed. With the phone to word context, the context of "POLITICAL /s/" is provided by the hybrid language model. Although the correct pronunciation of "/p/ /ah/ /l/ /ih/ /t/ /ah/ /k/ /ah/ /l/" is not on the best decoding path in terms of acoustic scores, the context of "POLITICAL /s/" from the language model will raise the total score of the sequence "/p/ /ah/ /l/ /ih/ /t/ /ah/ /k/ /ah/ /l/ /s/" so that it is higher than the sequence of "/p/ /l/ /ah/ /t/ /k/ /ah/ /l/ /s/". In this way, the missed keyword "POLITICAL" is recovered. In a similar vein, for the word to word context, the contributing context is the language model score of the word sequence "OF POLITICAL SCIENCE". All these contexts can be incorporated using different keyword spotting networks. Although they all use a monophone loop as the background model as shown in the lower path of Figure 1, these networks differ from each other by the upper path of the keyword graphs.

IV. Weighted Finite State Transducers for KWS

All the main components of a keyword spotting system can be represented as WFSTs as described below:

- Keyword spotting network W. This network is used to constrain the search space so that only keywords are produced and non-keywords are mapped as empty or some other tags. Figure 1 is a typical topology of a keyword spotting network W. The context information is also incorporated in W. For example, word to word transition, monophone to word transition and monophone to monophone transition in W are weighed using the probabilities provided by the language models.
- Language model G. The language model is used to provide the transition probabilities for W as the context information for the keywords. The language model must accommodate phone n-grams, word n-grams as well as the hybrid phone/word n-grams. In this paper, the hybrid word/phone language model is built using only the word n-gram counts to provide the phone to phone context, word to word context, and the hybrid word/phone context.
- Lexicon L. We use sub-word models for the keywords. Therefore, L is used to map the words in W and G to monophone sequences according to a lexicon.
- Acoustic Model H. The HMM topologies for all the sub-word models are encoded in H.

The composition operation [6] of the WFST can be used to combine the individual components. Meanwhile, optimisations such as determinisation and minimisation can reduce the network size dramatically thus make the keyword spotting quite efficient.

V. Context Dependent Keyword Spotting Networks with Language Model Contexts

In this section, we will elaborate how the language model contexts can be incorporated to the keyword spotting network. Depending on different contexts, three keyword spotting networks W are investigated, namely, the keyword loop, the

fixed length word context graph and the arbitrary length word context graph.

The context independent network W which is widely used in the keyword spotting literature is given in Figure 2. It is a loop of both keywords and the background monophones. The keyword spotting depends only on the acoustic scores alone to choose between the keywords and the background models.

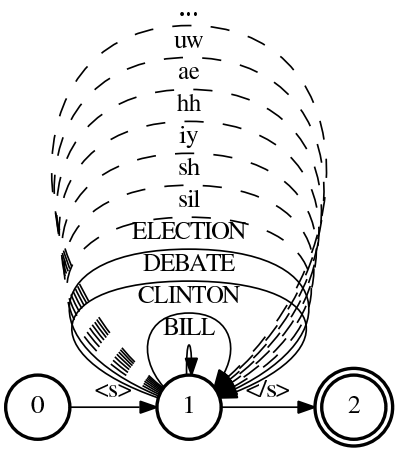

Fig. 2. Context independent keyword spotting graph. There is no word language model context used. The keyword spotting depends solely on the acoustic model. The dashed transitions denote the background model.

A. Keyword loop network: phone to word context

A straight-forward way of incorporating the contexts for the context independent graphs in Figure 2 is by weighting the transitions between the background monophone loops and the keywords. The filler is represented as monophone loops. The context information used here is the transition probability between the monophone fillers and the keywords obtained from the language model G. The incorporation of the language model can be realised by a composition of G with the context independent network W in Figure 2. It is important to note that the language model G will need to consider phone n-grams, word n-grams as well as the hybrid phone/word n-grams. The composition of $G \circ W$ is a WFST with monophone and keyword as both input and output. The monophone/keyword sequence is referred to as "hybrid sequence" in the following discussions. The lexicon WFST L is a mapping from keyword to its pronunciations. To accommodate the monophone loops, the lexicon also contains dummy entries mapping from monophones to themselves by viewing the monophones as a special case of "words" with a single phone in their pronunciations. By composing L with $G \circ W$, we have a WFST mapping from monophone sequences to hybrid sequences. A further composition with the acoustic model H provides the final network $H \circ L \circ G \circ W$ for the keyword spotting task.

B. Fixed length word context network: fixed length word to word context

Two filler models, left and right, are introduced in the word graph as shown in Figure 3 to provide the immediate left and right contexts for the middle keyword with a fixed context length of three. The filler models contain all the non-keywords in the word list. In addition to the two fillers, a monophone loop is used as the background model to compete with the keyword and filler model. The language model G is used to provide the transition probabilities between the background monophones to words, as well as the transitions between the word fillers and the keywords. The composition of $G \circ W$ is a keyword spotting graph with language model probabilities mapping from all the hybrid sequences to keywords and non-keyword tags. A further composition with the lexicon model $L \circ G \circ W$ provides a mapping from the monophone sequences to the keywords and non-keyword tags. Finally, $H \circ (L \circ G \circ W)$ transduces from monophone state indices to keyword and non-keyword tags. It is then used together with the acoustic model to perform keyword spotting.

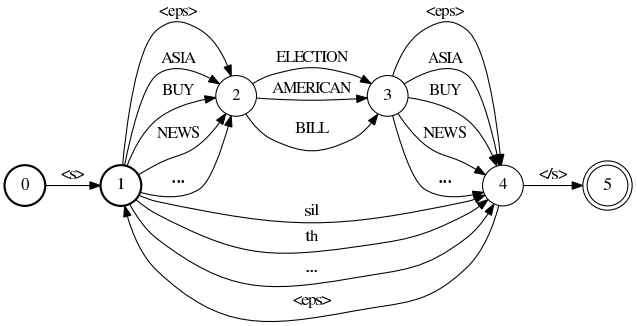

Fig. 3. Word graph. Two fillers (node 1 to 2 and node 3 to 4) together with the language model scores are used to provide the contexts for the keywords (node 2 to 3) in the upper portion of the graph. The lower portion is a monophone loop as the background model to speed up the spotting. The context length is three.

C. Arbitrary length word context network: arbitrary length word to word context

Only the immediate left and right contexts are used in Figure 3. As a generalisation, arbitrary word length contexts can be incorporated as shown in Figure 4. Similarly, the background model is a monophone loop. Arbitrary context length in Figure 4 is obtained through the left and right non-keyword loop fillers denoted as node 1 and 2. They are potentially more powerful than the fixed-length contexts. However, this may come at the cost of a potentially slower decoding speed since the search can ignore the background model and do a full decoding using only the upper portion of the decoding graph. The final decoding network $H \circ (L \circ G \circ W)$ is obtained same as the fixed length context network.

VI. The Hybrid Phone and Word Language Model

The previous section presents three different approaches of incorporating language model contexts. The phone to word

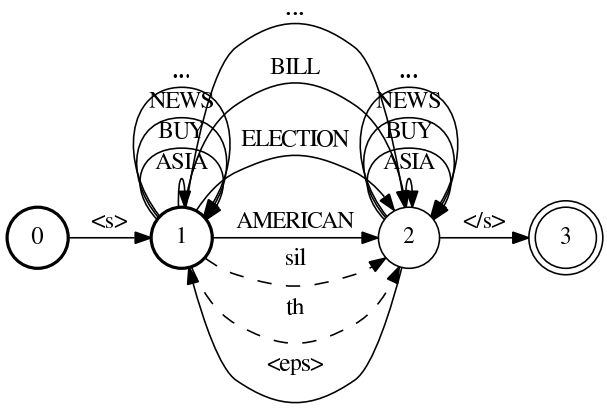

Fig. 4. arbitrary length word to word contexts. The two non-keyword loops denoted as node 1 and 2 are the contexts used to detect the middle keywords. The dashed transitions are the background monophone loops.

context is used through a keyword loop network with the background monophone loop. The context is accommodated by the phone to word language model scores. The word to word contexts are incorporated through the left and right word fillers with the word language model scores. In other words, the language model G not only needs to provide the word to word scores, but also the phone to word scores. Therefore, a hybrid monophone and word language model G must be trained. However, there does not exist a principled way of estimating the hybrid sequence directly from the training data since the language model training corpus is usually word based. We then propose to estimate the hybrid language model based on the word n-gram counts.

We use SRILM [1] to build a bigram and trigram word language model using the Gigaword corpus and the TDT 3 [19] transcription. The bigram counts and trigram counts are kept so that the hybrid word and monophone counts can be accumulated to build the hybrid word/phone language model.

A. Bigram hybrid counts accumulation

To accumulate the bigram hybrid phone/word counts from the word bigram counts, there are two possible scenarios:

$$C(p, w_2) = \sum_{w_1 \in W_{e(p)}} C(w_1, w_2) \quad (3)$$

$$C(w_1, p) = \sum_{w_2 \in W_{s(p)}} C(w_1, w_2) \quad (4)$$

where $C(\cdot)$ is the count of the n-gram, $W_{e(p)}$ denotes a set of words whose pronunciation ends with phone p, $W_{s(p)}$ is a set of words whose pronunciation starts with phone p. The count of the word bigram "$w_1 w_2$" is then added to both $C(p, w_2)$ and $C(p, w_2)$ hybrid bigram counts. For example, the bigram count of the word bigram "POLITICAL SCIENCE" should be added to the bigram counts for the hybrid sequence "/l/ SCIENCE" and "POLITICAL /s/" according to equation 3 and equation 4 respectively.

[1] http://www.speech.sri.com/projects/srilm/download.html

B. Trigram hybrid counts accumulation

To accumulate the trigram hybrid phone/word counts, there exists six cases:

$$C(p_1, p_2, w_3) = \sum_{w_1 \in W_{e(p_1 p_2)}} C(w_1, w_3) + \sum_{w_1 \in W_{e(p_1)}} \sum_{w_2 \in W_{p_2}} C(w_1, w_2, w_3) \quad (5)$$

$$C(w_1, p_2, p_3) = \sum_{w_2 \in W_{s(p_2 p_3)}} C(w_1, w_2) + \sum_{w_2 \in W_{p_2}} \sum_{w_3 \in W_{s(p_3)}} C(w_1, w_2, w_3) \quad (6)$$

$$C(p_1, w_2, p_3) = \sum_{w_1 \in W_{e(p_1)}} \sum_{w_3 \in W_{s(p_3)}} C(w_1, w_2, w_3) \quad (7)$$

$$C(w_1, p_2, w_3) = \sum_{w_2 \in W_{p_2}} C(w_1, w_2, w_3) \quad (8)$$

$$C(w_1, w_2, p_3) = \sum_{w_3 \in W_{s(p_3)}} C(w_1, w_2, w_3) \quad (9)$$

$$C(p_1, w_2, w_3) = \sum_{w_1 \in W_{e(p_1)}} C(w_1, w_2, w_3) \quad (10)$$

The notation is similar to the bigram case, where W_{p_2} denotes a set of words whose pronunciations contains only one single phone p_2 in the lexicon, $W_{s(p_1 p_2)}$ is a set of words whose first two phones in its pronunciation are $p_1 p_2$. Similarly, $W_{e(p_2 p_3)}$ is a set of words whose pronunciations ends with $p_2 p_3$. Given a word trigram "PRESIDENT BILL CLINTON", its count should be accumulated to the hybrid count of "/n/ /t/ BILL" (equation 5), 'PRESIDENT /b/ /ih/" (equation 6), "/t/ BILL /k/" (equation 7), "PRESIDENT BILL /k/" (equation 9), "/t/ BILL CLINTON" (equation 10). In addition to the previous 5 cases, the word trigram count of "OF A FEW" should be added to the hybrid trigram count of "OF /ah/ FEW" (equation 8).

C. Training the hybrid language model

Take note that the hybrid counts only have interleaved phone and word sequences after the accumulation. To include the word sequences, the original word bigram and trigram counts are also used. To accommodate the phone to phone transitions, the phone bigram and trigram counts are also used. The phone bigram and trigram counts are obtained from a phone level Gigaword corpus by expanding all the word tokens in the original Gigaword corpus to phone sequences according to a lexicon. With the three sets of counts, namely, the hybrid phone/word n-gram counts, the word n-gram counts and the phone n-gram counts, the hybrid language model can then be built. One may notice there is over-counting in this language model, since the phone counts and hybrid counts are related to the word counts. Moreover, the phone to phone counts and hybrid word/phone counts are usually much larger than the word to word counts, as the number of phones is significantly much smaller than the words. Therefore, scaling of the three language model scores is needed. In log domain, which is

adopted by the WFST framework, the scaling can be achieved by imposing the insertion penalties of the phone to phone and phone to word transitions.

VII. Experimental Results

In this section, the keyword spotting performance is reported for all three networks in terms of both accuracy and efficiency. The accuracy is evaluated using F-measure and the spotting speed using real-time (RT) factors. Take note that insertion penalties need to be tuned for various configurations in order to achieve the best performance. The results are all based on the best configuration after the tuning of the insertion penalties. The real-time factors are obtained by running the keyword spotting using the best configurations of all the three networks on the same machine. The machine has 8 G memory, 8 core Intel i7-2600 CPU @ 3.40GHz.

Context independent hybrid DNN/HMM is used as the acoustic model. The features are the standard 39-dimensional PLPs consisting of 13 static coefficients (12 PLP plus one C0 energy term) and the first and second derivatives. For the training of DNNs, up to five hidden layers with 1024 hidden units are trained. The input window size of the DNN input layer is 15 frames, rendering 585 input units. The output units are used to discriminate all the monophone states. The phone set has 40 phones and each phone is modelled as a 3-state left-to-right HMMs. Therefore, there are 120 output units. A baseline GMM/HMM system is needed to align the training data to get the training targets for the fine-tuning of the CI DNN. The baseline GMM/HMM system used has 4500 triphone state clusters and each state cluster is modelled with 20 Gaussian mixtures. Forced-alignment using the baseline model is performed on all the training data to provide the monophone state targets for the CI DNN.

The DNN model is trained with TNet [2] using GPUs on the 100 hours of the Topic Detection and Tracking - Phase 3 (TDT3) corpus [19]. The DNN training involves both pre-training and fine-tuning. The weights after the pre-training are used as the initial weights for fine-tuning the context independent DNN with 120 targets. The fine-tuning uses stochastic gradient descent (SGD) to minimise the cross-entropy between the labels and the network output. There are several important parameters for the pre-training of the DNN including the batch size, cache size, the learning rate for the Gaussian RBM layer, learning rate for the binary RBM layer, momentum and the weight cost. In all our experiments, these parameters are set as in Table II:

TABLE II
Parameter Settings for DNN Pre-training

batch size	cache size	momentum	weight cost
256	32768	0.0	0.0
learnrate gauss	iteration gauss	learnrate binary	iteration binary
0.001	100	0.08	50

[2]http://speech.fit.vutbr.cz/software/neural-network-trainer-tnet

TABLE III
Keyword Statistics

# Syllables	Keywords
1	BILL VOTE POOLS HOUSE STATES
2	CLINTON CAMPAIGN DEBATE REFORM COUNTRY OFFICE CONGRESS UNITED
3	PRESIDENT ELECTION GOVERNMENT WASHINGTON FEDERAL CANDIDATE SENATOR
4	AMERICAN POLITICAL REPUBLICAN DEMOCRATIC

The keyword spotting networks are implemented under the WFST framework using Kaldi [3]. The language models are trained with SRILM. The word language model is trained using the Gigaword English corpus interpolated with a language model trained on the TDT3 training transcription with a bigram perplexity of 286.307. The vocabulary size used for the language model training is 58K. The full decoding word error rate on the Hub4-97 set with the bigram word language model is 23.74% on the F0 portion of 177,432 frames lasting roughly half an hour of speech. The F0 portion of the Hub4-97 (LDC98S71, LDC98T28) broadcasting news evaluation set contains mainly the native, planned speech with a clean background.

The keywords are chosen from the Hub4-97 F0 portion related to the topic of "president election". There are totally 303 keyword tokens in the F0 evaluation set. The number of syllables of the keywords vary from 1 to 4. The number of tokens for the keyword with arbitrary syllable lengths is 37, 109, 114, 48 respectively as shown in Table III:

The spotting accuracy is reported using F-measure expressed as:

$$\text{Recall} = \frac{\#\text{keywords retrieved}}{\#\text{keywords}} \tag{11}$$

$$\text{Precision} = \frac{\#\text{keywords retrieved}}{\#\text{tokens retrieved}} \tag{12}$$

$$\text{F} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{13}$$

The performance is reported according to the number of the syllables of the keywords.

A. Baseline: context independent KWS

We first evaluate the KWS performance without any context as in the current KWS literature. The graph used is shown in Figure 2. There is no language model information involved, the KWS depends solely on the acoustic models. The performance is reported in Table IV:

The context independent KWS is able to achieve 221 hits out of the 303 tokens. Note the RT factor is only 0.13, which means it takes only 7.8 seconds to process a one-minute long utterance. The small RT factor should attribute to the WFST representation of the keyword spotting graph.

[3]http://kaldi.sourceforge.net/

TABLE IV
KEYWORD SPOTTING PERFORMANCE WITHOUT CONTEXTS

#Syllables	Recall	Precision	F-value	Hit	RT
1	0.78	0.53	0.63	29	-
2	0.69	0.74	0.71	75	-
3	0.73	0.8	0.77	80	-
4	0.77	0.65	0.73	37	-
Overall	0.73	0.72	0.72	221	0.13

B. KWS with phone to keyword context

The keyword spotting performance with the phone to keyword contexts from the hybrid language model is shown in Table V:

TABLE V
KWS WITH PHONE TO KEYWORD CONTEXTS

#Syllables	Recall	Precision	F-value	Hit	RT
1	0.65	1.0	0.79	24	-
2	0.69	0.8	0.74	75	-
3	0.81	0.73	0.77	88	-
4	0.9	0.62	0.74	43	-
Overall	0.76	0.75	0.75	230	0.13

Compared to the baseline configuration without any contexts, 9 more tokens are successfully retrieved. Therefore, there are 9 less missed keywords and 10 less false alarms.

Comparing Table V and IV, we can see that adding the phone to keyword contexts helps the detection of the long words with 3 or 4 syllables. Without any context information, the keyword spotting depends solely on the acoustic model which is not always reliable. Therefore, there are high chances that one of the phones may be miss-recognised thus resulting in a missed keyword. However, with the help of the hybrid language model, these recognition errors may be corrected since these wrongly recognised words may have a very low language model score thus their total scores are suppressed during the search. For example, the keyword "POLITICAL" is missed 5 times in the baseline. After incorporating the contexts, 4 of them are recovered. In an utterance, the keyword is recognised as the phone sequence of "/p/ /l/ /ah/ /t/ /k/ /ah/ /l/" where /ih/ is missed after /t/ rendering the keyword to be missed for the baseline. Due to the context of "POLITICAL s" which comes from the word sequence "POLITICAL SCIENCE", the keyword "POLITICAL" is recovered. Another example is that the word sequence "THE STATE HOUSE" is recognised as "THUS TAY HASS" because the phone /aw/ in "HOUSE" is wrongly recognised as /ae/ resulting in the keyword "STATE" and "HOUSE" to be missed. All these misses are corrected after incorporating the language model information. Efficiency wise, adding the phone to word language model context does not sacrifice the keyword spotting speed, the RT factor is the same as the baseline of 0.13.

C. KWS with fixed length word contexts

As shown in the last section, the incorporation of the phone to word contexts improves the keyword spotting accuracy compare to the baseline. In this section, more complex contexts are used. The keyword spotting graph is shown in Figure 3. Two word fillers are used as the left and right contexts for the keywords. The length of the context expansion is three as only the immediate left and right words are used for the detection of a keyword. The keyword spotting performance is given in Table VI:

TABLE VI
KWS PERFORMANCE OF FIXED LENGTH WORD CONTEXT

#Syllables	Recall	Precision	F-value	Hit	RT
1	0.59	0.88	0.71	22	-
2	0.79	0.84	0.82	87	-
3	0.8	0.81	0.75	87	-
4	0.92	0.64	0.75	44	-
Overall	0.79	0.79	0.79	240	0.25

Compared to the phone to word contexts in Table V, the fixed length word contexts recover 10 more tokens. The improvement mainly comes from the two-syllable words. The keyword "REFORM" is missed 7 times with the phone to keyword contexts. The number is reduced to only 1 with the fixed length word context. In one utterance, "REFORM" is wrongly recognised as "/er/ /f/ /ao/ /r/ /b/" because of the coarticulation influence of the previous word "FINANCE" and the following word "PROPOSAL". This may indicate the phone to word context alone is not robust enough. On the other hand, the word "FINANCE" AND "PROPOSAL" are used as the word contexts for the keyword "REFORM". Therefore, the keyword is recovered with the fixed length word context provided by the language model score. The detection speed is two times slower compared to both the baseline and the phone to word context. This is expected since the word context is significantly more complex than the phone to word context and the resulting spotting graph is also much larger. Nevertheless, the spotting speed is still very fast with a RT factor of 0.25 only.

D. KWS with arbitrary length word contexts

To further increase the context complexity, the spotting graph in Figure 4 is used to accommodate an arbitrary number of contexts of the keyword by using two word loops as fillers. The graph is even more complex than the fixed length word contexts configuration. On one hand, the graph has the potential of a full decoding if the search goes through only the upper portion of the graph without visiting the lower monophone garbage model. On the other hand, the decoding can also be as efficient as the one in Figure 2 if only the monophone loop is used to search for all the non-keyword words. Therefore, with the network in Figure 4, we have the flexibility of achieving a satisfactory spotting performance while enjoying a high spotting speed. The balance of this trade-off can be obtained from the scaling of the word language model score and the monophone language model score. Recall from section VI-C that scaling is needed for the hybrid language model. The

scaling can be achieved by imposing an insertion penalty on the monophone loop transitions.

From the word graph shown in Figure 4, if the lower portion of the monophone loop is removed, the graph becomes a full decoding graph. This serves as a reference point for the keyword spotting system in terms of both the spotting speed and the accuracy. The keyword spotting speed should not fall behind this reference point; otherwise, there is no need of performing the spotting task, i.e., we just need to wait for the full decoding to finish.

The performance of the full decoding is given in Table VII:

TABLE VII
KWS WITH FULL DECODING

#Syllables	Recall	Precision	F-value	Hit	RT
1	0.89	0.94	0.92	33	-
2	0.91	0.94	0.92	99	-
3	0.84	0.94	0.88	91	-
4	0.98	0.96	0.97	47	-
Overall	0.89	0.94	0.92	270	0.9

As expected, the full decoding has a much better spotting accuracy but with a much slower decoding speed. The full decoding accuracy is the upper bound for our keyword spotting systems. The RT factor of the full decoding should serve as a lower bound. The decoding graph for the varying length word context is more than 4 Gigabytes because of the the left and right word loop fillers, which is almost 2 times larger than the full decoding graph and the one in the fixed length word context. Decoding with the graph with such a large size graph almost consumes all the memories and causes a lot of IO operations which is not desirable. Therefore, we prune the decoding graph so that the resulting graph is the same size as the full decoding and the fixed length word context. The pruning threshold is chosen as 29 which means all the paths with the weight larger than the best path by 29 are pruned. In addition, tuning of the insertion penalty is needed to achieve a balanced trade-off between the spotting accuracy and efficiency. We choose the insertion penalty of -0.5 as it has the best trade-off between the efficiency and the accuracy. The detailed performance is reported in Table VIII.

TABLE VIII
KWS WITH ARBITRARY LENGTH WORD CONTEXTS

#Syllables	Recall	Precision	F-value	Hit	RT
1	0.76	0.93	0.88	29	-
2	0.84	0.92	0.88	92	-
3	0.81	0.96	0.88	88	-
4	0.96	0.91	0.93	46	-
Overall	0.84	0.93	0.89	255	0.29

Compared to the full decoding, 15 more keywords are missed. However, the spotting efficiency is greatly improved with a RT factor of only 0.29 comparable to the full decoding of RT 0.9. In other words, the speed is three times faster than the full decoding. In addition, we also attempt to adjust the search beams of the full decoding so that it has the same RT factor as the arbitrary length contexts configuration, the F-value of the full decoding with RT factor of 0.29 is 0.85. Compared to the arbitrary length contexts configuration, 21 more tokens are missed. This clearly shows the advantage of the arbitrary length contexts over the full decoding. Compared to the fixed length word contexts, the arbitrary length contexts configuration offers a significant performance boost: 15 more tokens are recovered. In one utterance, the keyword "BILL" is missed with the fixed length word contexts because it is wrongly recognised as "/b/ /ah/". The immediate left and right contexts of the keyword are "DAY" and "CLINTON'S". Obviously the right context helps much more than the left context. However, even with these contexts, "BILL" is still not recovered. If we allow more contexts in the arbitrary length word contexts configuration, the keyword "BILL" can be successfully recovered as "RECENT ELECTION DAY BILL CLINTON'S" is a much stronger context than "DAY BILL CLINTON'S".

E. KWS performance comparison of various contexts

The performance of the various keyword spotting networks are summarised in Table IX in terms of both accuracy (F-values) and efficiency (RT-factors).

TABLE IX
PERFORMANCE COMPARISON OF VARIOUS CD KWS NETWORKS

	Baseline	Keyword loop	Word fillers	Word loop fillers	Full decoding
RT factors	0.13	0.13	0.25	0.29	0.9
F-values	0.73	0.75	0.79	0.89	0.92

The baseline context independent network has the fastest speed with a RT factor of 0.13. Adding the phone to word contexts using a keyword loop has the same RT factor as the baseline with a better spotting accuracy. Incorporating the fixed length word contexts increases the network size dramatically compared to the keyword loop. Therefore, it has a slower decoding speed which is only half of the baseline and the keyword loop network. The arbitrary length word context using the word loop fillers has an even larger network thus decreases the spotting speed further. However, the RT factor difference between the fixed and arbitrary word length contexts is quite small (0.25 vs 0.29). The full decoding has the worst RT factor of 0.9 which is almost three times slower than the word contexts and 7 times slower than the baseline. Accuracy wise, compared to the baseline context independent network, adding the language model context information is essential for better spotting accuracies. The arbitrary length word context has the best accuracy without sacrificing the speed too much compared to the fixed length word context. The performance is comparable to the full decoding but with a three times faster decoding speed.

VIII. Error Analysis

In this section, we will examine the errors for both the word graph with and without the language model information. The context independent graph in Figure 2 is used as the baseline configuration without any contexts. The best configuration of the arbitrary length word contexts in Table VIII is used to demonstrate the advantages of the incorporation of the language model contexts. To do this, a forced-alignment is performed on the Hub4-97 F0 portion with a well trained triphone GMM/HMM model. The forced-alignment gives the time information for each occurrence of the keywords as the ground truth. The keyword spotting results are then aligned with the ground truth to get the types of errors. If the difference of the middle points between a putative keyword occurrence and the ground truth falls within a threshold number of frames, the putative hit is deemed to be correct. If no keyword is in the ground truth within the putative time interval, the putative hit is a false alarm. If the spotter failed to detect one keyword in the ground truth, the keyword is treated as a deletion error.

If no language model is used to provide the context information, the spotting depends solely on the acoustic model scores. It is well known that the acoustic scores are not always reliable. For the keyword spotting task, the unreliable acoustic scores can cause even a more severe problem: a single wrongly recognised phone will cause the whole word to miss, incurring a deletion error. With the language model contexts, this problem can be greatly circumvented. Examples are given in the previous experiment sections.

As for insertion errors, there are mainly five types:

Type 1: Derived words. For example, "AMERICAN" can be easily recognised as "AMERICA", "VOTE" as "VOTES", etc.

Type 2: The keyword is part of a longer word. For example, the word "COUNTRY" is the same as the "CONTRI" in the word "CONTRIBUTION", "ELECTION" is part of "SELECTION", "BILL" in "BILLION".

Type 3: The keyword is acoustically similar to another word. The keyword "POOLS" sounds very similar to "LOOPHOLES".

Type 4: The keyword spans multiple short words. The word "OFFICE" is wrongly inserted in place of the word sequence "OFF ITS".

Type 5: All the remaining errors due to some serious recognition errors. For example, "AMERICAN" is inserted in place of the word sequence "OUT ASSAULT".

The five types of insertion errors for each CD KWS system is given in Table X:

TABLE X
Error types of various CD KWS networks

Error Type	1	2	3	4	5	Total
Context Independent	57	13	6	1	16	93
Keyword Loop	64	4	4	1	8	81
Word fillers	42	3	3	0	8	64
Word loop fillers	13	0	2	0	8	23

From the table, we can see the performance gain of the keyword loop network over the baseline attributes to the recovery of type 2, type 3 and type 5 errors although it has a larger value of type 1 errors. The fixed word length context in the word filler network helps the correction of the type 1, type 2, and type 3 errors. For the arbitrary length word context in the word loop filler network, all types of errors of the baseline system are corrected to a significant amount.

Type 1 error is the most common insertion error which has 57 instances out of the total 93 insertion errors of the baseline graph without contexts. The arbitrary length word context graph reduces the insertion errors from 93 to 23. The improvement mainly comes from the recovery of the case 1 errors. The reason is that the derived words of the type 1 errors are mainly because of the plural or adjective forms of a keyword stem. With the context information in the language model, these derived forms can be easily distinguished from the stem. For example, the keyword "PRESIDENT" is inserted in place of all occurrences of its adjective form "PRESIDENTIAL", rendering 13 insertions. With the language information, 12 of them are resolved. Another example is the insertion of the keyword "REPUBLICAN" in place of "REPUBLICANS". There are 5 such errors in the baseline configuration. With the arbitrary length word contexts, all of them are recovered. From the above analysis, we can see that adding the language model information to a word graph can significantly reduce both the high insertion errors from the phone graph and the deletion errors from the word graph without language model.

IX. Conclusion

The incorporation of language model into keyword spotting to provide contexts for the keywords is investigated for three different context configurations: phone to keyword contexts, fixed length word contexts and arbitrary length word contexts. A hybrid language model with both words and phones tokens are trained on the word n-gram counts to provide all the contexts. A spotting graph is designed for each of these context configurations: keyword loop graph for the phone to keyword contexts, left and right word fillers for the fixed length word contexts and a left and right word loop fillers for the arbitrary length word contexts. Compared to the baseline configuration without any language model contexts, the keyword spotting performance is improved by the introduction of the language model for all the context configurations. Among the three context dependent keyword spotting graphs, the one with the fixed length word context outperforms the keyword loop context in terms of accuracy. As a generalisation of the fixed length word context, the arbitrary length word context provides the best performance among the three context dependent keyword spotting configurations. Efficiency wise, the keyword loop context has the same RT factor as the baseline graph without any context. It is also twice faster than the fixed length word contexts and the arbitrary length word context. Compared to the full decoding, the arbitrary length word context has a comparable spotting performance but has a spotting speed which is three times faster.

Future work can be extended to see whether the context information provided by the language model can help the spotting for informal speeches, or under noisy conditions.

ACKNOWLEDGMENT

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

REFERENCES

[1] J. R. Rohlicek, P. Jeanrenaud, K. Ng, H. Gish, B. Musicus, and M. Siu, "Phonetic training and language modeling for word spotting," 1993, ICASSP'93, pp. 459–462, IEEE Computer Society.

[2] P. Jeanrenaud, M Siu, J.R. Rohlicek, M. Meteer, and H. Gish, "Spotting events in continuous speech," in *In ICASSP*, 1994.

[3] Yong Ling, *Keyword spotting in continuous speech utterances*, Master Thesis, McGrill University, 1999.

[4] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 2006, 2006.

[5] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Trans. Audio, Speech and Lang. Proc.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[6] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Weighted Finite-State Transducers in Speech Recognition," 2000.

[7] R.W. Christiansen and C.K. Rushforth, "Detecting and locating key words in continuous speech using linear predictive coding," *IEEE Trans. Acoustic, Speech, Signal Processing*, vol. 25, no. 5, pp. 361–367, 1977.

[8] A. Higgins and R. Wohlford, "Keyword recognition using template concatenation," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.*, 1985, vol. 10, pp. 1233–1236.

[9] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, 1989, pp. 627–630.

[10] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, vol. 38, no. 11, pp. 1870–1878, 1990.

[11] H. Bourlard, B. D'Hoore, and J.-M. Boite, "optimizing recognition and rejection performance in word spotting systems," in *In ICASSP*, 1994.

[12] R.C. Rose, "Keyword detection in conversational speech utterances using hidden markov model based continuous speech recognition," *Computer Speech & Language*, vol. 9, no. 4, pp. 309 – 333, 1995.

[13] I Motlicek, Petr; Valente, Fabio; Szöke, "Improving acoustic based keyword spotting using LVCSR lattices," in *ICASSP*, 2012, pp. 4413–4416.

[14] Igor Szoke, Petr Schwarz, Pavel Matejka, and Martin Karafiat, "Comparison of keyword spotting approaches for informal continuous speech," in *In Proceedings Eurospeech*, 2005.

[15] Herve A. Bourlard and Nelson Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[16] T. Zeppenfeld and A. Waibel, "A hybrid neural network, dynamic programming word spotter," in *ICASSP*, 1992, pp. 77–80.

[17] Martin Wollmer, Bjorn Schuller, and Gerhard Rigoll, "Keyword spotting exploiting long short-term memory," *Speech Communication*, vol. 55, no. 2, pp. 252 – 265, 2013.

[18] Qing Guo, Yonghong Yan, Zhiwei Lin, Baosheng Yuan, Qingwei Zhao, and Jian Liu, "Keyword spotting in auto-attendant system," in *INTERSPEECH*, 2000, pp. 1050–1052.

[19] David Graff, Chris Cieri, Stephanie Strassel, and Nii Martey, "The TDT-3 Text And Speech Corpus," in *in Proceedings of DARPA Broadcast News Workshop*. 1999, pp. 57–60, Morgan Kaufmann.