A Neural Network Keyword Search System for Telephone Speech

Kevin Kilgour and Alex Waibel

Karlsruhe Institute of Technology, IAR, Interactive Systems Lab
Adenauerring 2, 76131 Karlsruhe
http://www.kit.edu
kevin.kilgour, alex.waibel@kit.edu

Abstract. In this paper we propose a pure *"neural network"* (NN) based keyword search system developed in the IARPA Babel program for conversational telephone speech. Using a common keyword search evaluation metric, *"actual term weighted value"* (ATWV), we demonstrate that our NN-keyword search system can achieve a performance similar to a comparible but more complex and slower *"hybrid deep neural network - hidden markov model"* (DNN-HMM Hybrid) based speech recognition system without using either an HMM decoder or a language model.

Keywords: ATWV, Deep Neural Networks, Keyword Search

1 Introduction

Keyword search is a spoken language processing task in which the goal is to find all occurrences of a keyword (one or more written words) in a large audio corpus. It is sometimes also referred to as spoken term detection. In this paper we'll focus on two phase keyword search systems, where in the first phase the audio corpus in processed and indexed without any knowledge of the keywords. After the keyword list becomes known this index is queried and a prediction list is rapidly returned without re-processing the audio data.

All modern high performing keyword search systems are based on speech recognition systems [10,2,7] or on a combination of multiple speech recognition systems [9]. This requires not only an acoustic model and a language model but also that the audio has to be processed by a speech recognition decoder, which can be quite time consuming. While some work has been performed on isolated keyword detection [12] without the use of a speech recognition system, their setups generally require knowledge of the keywords before training the system.

We propose an alternative system that only uses a neural network of a complexity similar to an acoustic model and directly produces a indexable set of keyword predictions with confidences from the audio corpus. In section 2 of this paper we formally describe the keyword search task and describe how the performance of a keyword search system can be measured. After explaining the baseline speech recognition based keyword search system in section 3, we present our neural network keyword search system in section 4 and show how its output can be easily converted into an indexable format.

A. Ronzhin et al. (Eds.): SPECOM 2014, LNAI 8773, pp. 58–65, 2014.

Table 1. Overview of the provided data for the 80h Vietnamese full LP and the 10h Lao limited LP

Language Code	Name	Version	language pack	dictionary size	transcribed text
107	Vietnamese	IARPA-babel107b-v0.7	full LP	6857	∼120.000
203	Lao	IARPA-babel203b-v3.1	limited LP	4215	∼98000

The evaluation of both keyword search systems is presented and discussed in section 5 while section 6 contains a short summary and conclusion.

2 Keyword Search

In this section we'll present an overview of the keyword search task with particular focus on the IARPA Babel program, which aims to design keyword search systems *that can be rapidly applied to any human language in order to provide effective search capability for analysts to efficiently process massive amounts of real-world recorded speech* [8].

2.1 Data Overview

In order to evaluate the rapid application to new languages the Babel program concentrates on under-resourced languages like Tagalog, Vietnamese or Lao. Keyword search systems can only be trained on the provided 80 hours of transcribed audio data in the full language pack (LP) case or 10 hours in the limited LP case. A pronunciation dictionary for the transcribed words is also provided. We tested our setup on both the Vietnamese full LP case and on the Lao limited LP case (see table 1). These languages are chosen because of their low vocabularies (<10.000).

2.2 Actual Term Weighted Value (ATWV)

Given an audio test set (T seconds long) and a list of N keywords, a keyword search (KWS) system should produce a list of keyword occurrences, with timestamps, for each keyword. In this context the keywords do not necessary have to be single words and can instead be bi-grams or short phrases like *"the 8 o'clock bus"* or *"running rabbit"*. When the list of predicted keywords is compared to the reference, all predictions within 0.5 seconds of a reference keyword occurrences are counted as hits ($N_{\texttt{corr}}$) and the other incorrect predictions are counted as false alarms ($N_{\texttt{FA}}$). The total occurrences of a keyword in the reference is referred to as $N_{\texttt{true}}$ and all keywords not detected by the keyword search system are referred to as *misses*.

The *"actual term weighted value"* (ATWV) is computed over all N keywords that occur at least once in the reference.

$$ATWV = \frac{1}{N}\sum_{w}\left[\frac{N_{\texttt{corr}}(w)}{N_{\texttt{true}}(w)} - \beta\frac{N_{\texttt{FA}}(w)}{T - N_{\texttt{true}}(w)}\right]$$

The balancing factor β weighs the importance of false alarms compared to misses. In the Babel program β is typically set to 999.9 and T, the length of the dev data, is about

36000s. The initial ATWV target is set at 0.3 (often written as 30%). Since $T \gg N_{\texttt{true}}$ we can simplify the ATWV definition to:

$$ATWV \approxeq \frac{1}{N} \sum_{w} \left[\frac{N_{\texttt{corr}}(w)}{N_{\texttt{true}}(w)} - \frac{N_{\texttt{FA}}(w)}{36} \right]$$

which shows us that while the penalty for a false alarm is constant at roughly $\frac{1}{36N}$, the penalty for a miss depends on the number of true occurrences of the keyword in the reference. Keywords that only occur once incur a miss penalty of $\frac{1}{N}$ (36 times its false alarm penalty), while keywords that occur 36 times have the same miss and false alarm penalty.

A KWS system that not only produces keyword predictions but also confidences associated with each prediction allows us to use keyword specific thresholds in order to minimize our expected penalty [9] .

$$thr(w) = \frac{\beta \cdot N_{\texttt{true}}(w)}{T + (\beta - 1) \cdot N_{\texttt{true}}(w)}$$

Since $N_{\texttt{true}}(w)$ is not known, it has to be estimated based on known information. A simple and effective method involves using the sum of the detection confidences $S(w)$.

$$\hat{N}_{\texttt{true}}(w) = \alpha \cdot S(w) = \sum_{d \in \texttt{detections}(w)} \texttt{conf}(d)$$

where α is a boosting factor that compensates for the fact that not all occurrences of the keyword will be present in the prediction list [17].

As an input to our indexing and keyword search tool [11] we use confusion networks that are normally generated by speech recognition systems.

3 Keyword Search using Speech Recognition

The best performing keyword search systems are based on automatic speech recognition systems (ASR) which, instead of being tuned to minimize word error rate, are tuned to generate a confusion network (or word lattice) that maximizes ATWV (see figure 1). As a baseline we use a hybrid DNN-HMM ASR system with a modular DNN acoustic model [3] and a 3gram Kneser-Ney smoothed language model [5]. The language model is trained on the audio transcripts.

Since both Vietnamese and Lao are tonal languages we decided to use pitch features [13] as well as the normal log MEL features. In the following paragraphs we describe the Vietnamese acoustic model in detail since we use its topology as the basis for our neural network keyword search system. For Lao limited LP system we used the best acoustic model that we had available.

We use 40 log MEL features augmented with the 14 pitch features as input features to the DNN acoustic model which is trained in two stages. First a normal bottleneck feature network [4] with 5 hidden layers prior to the bottleneck and 1600 neurons in

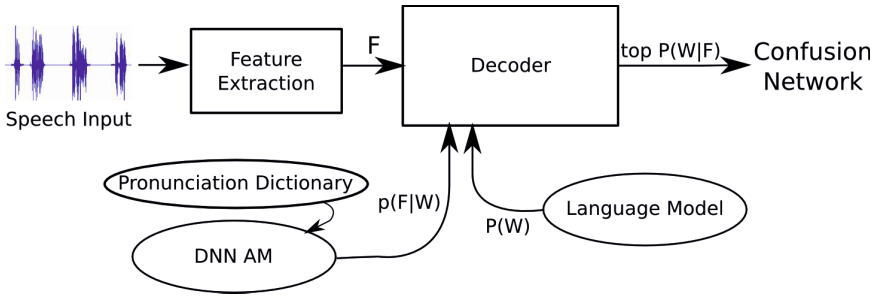

Fig. 1. Overview of a normal speech recognition based keyword search system. The confusion networks generated by the speech recognizer can by indexed and queried a later point in time.

each layer is trained on a window of 13 input features (10ms frameshift). A final hidden layer and the target layer are discarded after training. The target layer consists of 10000 neurons each corresponding to a context dependent phone-state and the bottleneck contains 42 neurons. This network can be seen as a non-linear dimensionality reduction of the 702 (13*54) dimensional input feature vector to a 42 dimensional bottleneck feature (BNF).

A 15 frame window (also 10ms frameshift) of these BNFs is used as the input to the second stage which consists of a further 5 hidden layers with 1600 neurons each and a 10000 neuron final target layer where again each neuron corresponds to a context dependent phone-state. All layers use a sigmoid activation function, except for the output layer which uses the softmax activation function.

Both stages are pre-trained layer-wise using denoising auto encoders [15] for 1 million mini-batches per layer (mini-batch size 256) with constant learning rate of 0.01. For fine-tuning the newbob learning rate schedule is used which starts of with a constant learning rate of 1 until that no longer improves the cross-validation accuracy after which the learning rate is exponentially decayed. Our training setup utilizes the Theano library [1].

Prior to decoding the audio data is segmented into utterances by using the Gaussian Mixture Models (GMM) method proposed by [6]. The decoding is performed using the IBIS single pass decoder of the Janus Recognition Toolkit (JRTk) [14] resulting in set of confusion networks from which we can generate the list of keyword predictions with confidences. After applying the keyword specific thresholds described at the end of section 2, we score the ATWV.

4 Neural Network Keyword Search System

The neural network keyword search system uses the same 40 logMEL + 14 pitch features as the speech recognition system. The topology of the neural network shown in figure 2 is almost identical to the topology of the DNN acoustic model of the speech recognition system. The bottleneck feature part of the network is the same but the BNFs are stacked over a larger window of 29 frames and the final layer contains only 6857 neurons (Lao 4215), each associated with either a word, a noise or silence. This output layer gives the occurrence probabilities for each of the 6857 known words.

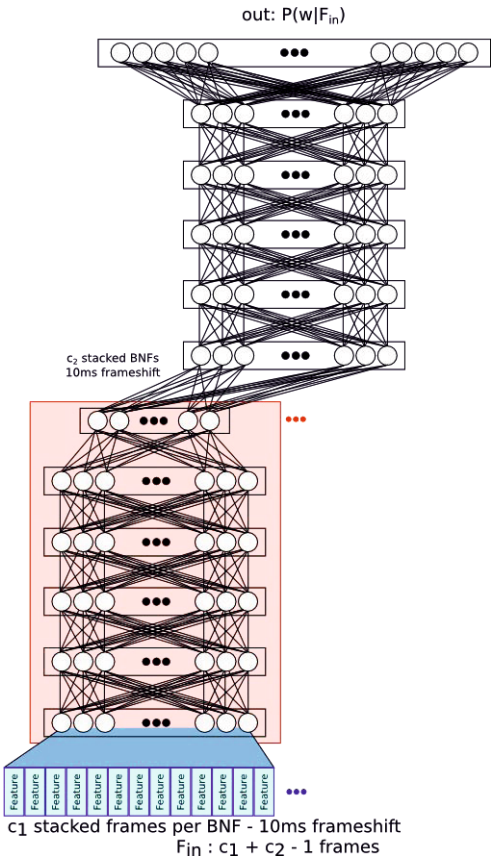

Fig. 2. Neural Network Topology; the neural network consists of two subnetworks. The output of the initial DBNF network from a standard ASR system is stacked over c_2 frames and used as the input of the deep word classification network. The input feature vector to the DBNF consists of 40 log MEL features concatenated with 14 pitch features and stacked over c_1 frames. All hidden layers contain 1600 neuron except for the bottleneck layer (42) and the target layer (6800).

Training is also performed in an similar manner to the training of the DNN acoustic model with exception that a smaller initial learning rate of 0.5 is used in final fine tuning stage.

Processing the audio with the NN keyword search system involves first segmenting it into into utterances using either the GMM or the SVM method of [6] and then extracting the required features from the audio which can be passed through the neural network.

Using a frame shift of 10ms results in a our neural network generating 100 6857 dimensional probability vectors per second of audio. The fact that some utterances can be over 10 seconds forces us to deal with a large amount of data. We collect all the probability vectors of an utterances into word frame probability matrix (WFPM) where each row j represents one frame and each column i corresponds to a word.

Table 2. ATWV results and real time factors (rtf) of a standard speech recognition based keywords search systems and our proposed neural network based keyword search system on the Babel Vietnamese and Lao development sets

Language	ASR KW	NN KWS
Vietnamese full LP	27.94% ATWV / 5.3 rtf	31.35% ATWV / 3.4 rtf
Lao limited LP	24.44% ATWV / 5.7 rtf	8.62% ATWV / 3.7 rtf

4.1 WFPM Post processing

In order to perform a keyword search using keyword specific thresholds with our existing tools we convert the generated WFPM to a confusion network-like structure. As an initial step the matrix is smoothed by averaging the word probabilities across multiple frames

$$[p_{i,j}]_{k-smooth} = \frac{1}{2k+1} \sum_{j-k}^{j+k} p_{i,j}$$

and converted into a sparse matrix by setting all probabilities below a given threshold to 0.

$$[p_{i,j}]_{c-filt} = \begin{cases} p_{i,j} & \text{if } p_{i,j} > c \\ 0 & \text{otherwise} \end{cases}$$

The resolution of the WFPM is reduced from one row every 10ms to one row every $x \cdot 10$ms by only keeping rows where $\left\lfloor \frac{x}{2} \right\rfloor \equiv j \mod x$. After this the rows can be re-normalized since we lost some probability mass when going to a sparse matrix. The sparse WFPM can now be converted into a confusion network be treating each row, j, as list a transition portabilities for their associated words between nodes $j-1$ and j.

This pseudo confusion network can be scored in the same way as the normal confusion network.

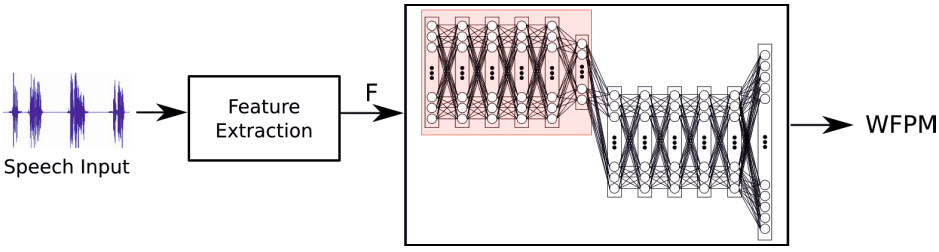

Fig. 3. Overview of the proposed neural network keyword search system

5 Evaluation

As evaluation data we used the Babel 107 Vietnamese developed set and the Babel 203 Lao development set. As can be seen in table 2 the neural network keyword search system slightly outperforms the ASR keyword search system on the Vietnamese full LP set but is far behind it on the Lao limited LP set. One reason for this is due to the fact that a relatively poor (ATWVs $>50\%$ have been reported on this set) Vietnamese baseline system was used in this setup; it was chosen because its acoustic model has a similar topology to our proposed neural network keyword search system. The poor performance of our NN keyword search system on the Leo limited LP set is in part due to the fact that we didn't tune it in towards the new language or towards the low resource condition.

The evaluation was performed on 4 16-core AMD Opteron 6276 CPUs where we measured the per-core real time factor. The fact that each Opteron only has 8 floating point units (1 for every 2 cores) reduces the real-time performance of both systems significantly. The matrix multiplication were performed with the openBLAS library [16].

6 Conclusion

This paper has introduced a neural network keyword search system based on the design of a modular DNN acoustic model used for speech recognition. On the Vietnamese full LP set our system is able to achieve a performance similar to that of a full speech recognition system and above the Babel initial goal of 30% ATWV. It also only requires $\frac{2}{3}$ the CPU time. Further work still needs to be carried out in three areas, closing the gap to current best ASR based keyword search systems ($> 50\%$ ATWV), investigating the problems with our limited LP system and dealing with the problems that occur with very large vocabularies. Until then this neural network keyword search system can only be recommended in very specific circumstances.

Acknowledgments. Supported in part by the Intelligence Advanced Research Projects Ac- tivity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

References

1. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: A CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy) (June 2010) Oral Presentation

2. Cui, J., Cui, X., Ramabhadran, B., Kim, J., Kingsbury, B., Mamou, J., Mangu, L., Picheny, M., Sainath, T.N., Sethy, A.: Developing speech recognition systems for corpus indexing under the iarpa babel program. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6753–6757. IEEE (2013)
3. Gehring, J., Lee, W., Kilgour, K., Lane, I., Miao, Y., Waibel, A., Campus, S.V.: Modular combination of deep neural networks for acoustic modeling. In: Proc. Interspeech, pp. 94–98 (2013)
4. Gehring, J., Miao, Y., Metze, F., Waibel, A.: Extracting deep bottleneck features using stacked auto-encoders. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3377–3381. IEEE (2013)
5. Goodman, J., Chen, S.: An empirical study of smoothing techniques for language modeling. Tech. rep., Technical Report TR-10-98, Harvard University (August 1998)
6. Heck, M., Mohr, C., Stüker, S., Müller, M., Kilgour, K., Gehring, J., Nguyen, Q.B., Van Nguyen, H., Waibel, A.: Segmentation of telephone speech based on speech and non-speech models. In: Železný, M., Habernal, I., Ronzhin, A. (eds.) SPECOM 2013. LNCS (LNAI), vol. 8113, pp. 286–293. Springer, Heidelberg (2013)
7. Hsiao, R., Ng, T., Grezl, F., Karakos, D., Tsakalidis, S., Nguyen, L., Schwartz, R.: Discriminative semi-supervised training for keyword search in low resource languages. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 440–445. IEEE (2013)
8. IARPA: Iarpa babel program - broad agency announcement, baa (2011), `http://www.iarpa.gov/Programs/ia/Babel/solicitation_babel.html`
9. Karakos, D., Schwartz, R., Tsakalidis, S., Zhang, L., Ranjan, S., Ng, T., Hsiao, R., Saikumar, G., Bulyko, I., Nguyen, L., et al.: Score normalization and system combination for improved keyword spotting. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 210–215. IEEE (2013)
10. Kingsbury, B., Cui, J., Cui, X., Gales, M.J., Knill, K., Mamou, J., Mangu, L., Nolden, D., Picheny, M., Ramabhadran, B., et al.: A high-performance cantonese keyword search system. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8277–8281. IEEE (2013)
11. Kolkhorst, H.: Strategies for Out-of-Vocabulary Words in Spoken Term Detection. Undergraduate thesis (2011)
12. Kurniawati, E., George, S.: Speaker dependent activation keyword detector based on gmm-ubm (2013)
13. Metze, F., Sheikh, Z.A., Waibel, A., Gehring, J., Kilgour, K., Nguyen, Q.B., Nguyen, V.H.: Models of tone for tonal and non-tonal languages. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 261–266. IEEE (2013)
14. Soltau, H., Metze, F., Fugen, C., Waibel, A.: A one-pass decoder based on polymorphic linguistic context assignment. In: IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2001, pp. 214–217. IEEE (2001)
15. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research 9999, 3371–3408 (2010)
16. Wang, Q., Zhang, X., Zhang, Y., Yi, Q.: Augem: Automatically generate high performance dense linear algebra kernels on x86 cpus. In: Proceedings of SC 2013: International Conference for High Performance Computing, Networking, Storage and Analysis, p. 25. ACM (2013)
17. Wang, Y.: An in-depth comparison of keyword specific thresholding and sum-to-one score normalization. Tech. rep., Technical Report, Carnegie Mellon University (2014)