INFORMATION RETRIEVAL METHODS FOR AUTOMATIC SPEECH RECOGNITION

Xiaoqiang Xiao[*]

Department of Electrical Engineering
The Pennsylvania State University
University Park, PA
xxx106@psu.edu

Jasha Droppo and Alex Acero

Speech Research Group
Microsoft Research, Redmond, WA
{jdroppo,alexac}@microsoft.com

ABSTRACT

In this paper, we use information retrieval (IR) techniques to improve a speech recognition (ASR) system. The potential benefits include improved speed, accuracy, and scalability. Where conventional HMM-based speech recognition systems decode words directly, our IR-based system first decodes subword units. These are then mapped to a target word by the IR system. In this decoupled system, the IR serves as a lightweight, data-driven pronunciation model. Our proposed method is evaluated in the Windows Live Search for Mobile (WLS4M) task, and our best system has 12% fewer errors than a comparable HMM classifier. We show that even using an inexpensive IR weighting scheme (TF-IDF) yields a 3% relative error rate reduction while maintaining all of the advantages of the IR approach.

Index Terms – Speech Recognition, Information Retrieval, Direct Modeling, Vector Space Model, Language Model.

1. INTRODUCTION

Hidden Markov models (HMM) have been used in automatic speech recognition (ASR) for the last 40 years. Although they are powerful modeling tools, HMMs have sequencing constraints associated with difficulties in modeling of and non-robustness to accented speech or background noise. Recently, there has been growing interest in direct modeling for speech recognition, directly maping acoustic features to word sequences. Some examples of direct modeling in speech recognition include maximum-entropy Markov models (MEMMs) [1], conditional random fields (CRFs) [2] and flat direct models [3].

The method proposed in this paper is inspired by the success of information retrieval for flexible language modeling in voice search applications [4], and falls into the category of direct model based ASR. In [4], information retrieval (IR) techniques are used to map from words to business listings. Here, we consider the acoustic units themselves as "terms" and the business listings as "documents," and use an IR system to map directly from acoustic units to the desired listings.

A similar idea of combining IR and ASR has been recently proposed as vFORMS in [5]. The vFORMS system requires two passes of ASR. The first pass uses lightweight recognition of sub-word units together with an IR engine to prune the search space into a short list of possible words. A second pass of word-based ASR, constrained to this short list of possibilities, can then easily find the best hypothesis. Instead of using two passes scheme, we propose a method only requires one pass that directly map the sequence of recognized sub-word units to the final hypothesis.

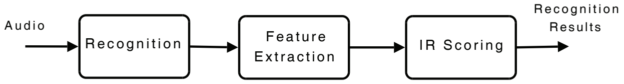

Figure 1. A block diagram of the proposed method.

There are many possible choices of acoustic unit such as acoustic features based on phonetic lattices [6] and sub-word units produced by phonetic recognition systems [7]. In this paper, we choose n-gram features extracted from phonetic recogntion, multiphone recognition and word recognition (described in section 2.1 and 2.2).

The advantage of using IR in mapping from acoustic units to listings is that it provides a more flexible pronunciation model. In modern ASR systems the dictionary is usually not trained. If the speaker has an accent, or talks casually, or if there is an abundance of background noise, there will be a mismatch between the expected pronunciation from the dictionary and the realized pronunciation of the utterance. Given enough training data, the IR system can replace a small number of canonical pronunciations with a learned, discriminative distribution over sub-word units for each listing.

PHARMACY (canonical pronunciation)	F AA R M AX S IY
PHARMACY (training pronunciation 1)	F AO R M AX S IY
PHARMACY (training pronunciation 2)	F AY R IH S IY
PHARMACY (training pronunciation 3)	F AY R N AX S IY

Table 1. Document combining canonical and training pronunciations.

Another advantage of using IR in ASR is that the vector space model used in IR has no sequencing constraints, which could lead to a system more robust to disfluencies and noise. Due to the discriminative nature exists of the IR engine, it should be possible to recognize a word by emphasizing a well pronounced discriminative core and de-emphasizing any noisy extremities. In the case of PHARMACY (Table 1), the first syllable may be more stable than the other two.

The remainder of this paper is organized as follows. The details of the proposed method are summarized in section 2. Experimental results and performance studies are provided in section 3. Section 4 concludes the paper.

2. METHODS

A block diagram of the proposed method is shown in Figure 1. The proposed speech recognition procedure is divided into three main tasks: (A) the recognition task, (B) the feature extraction task, and (C) the IR scoring task.

[*] This work is done in while the first author worked as an intern in Microsoft Research, Redmond.

978-1-4244-4296-6/10/$25.00 ©2010 IEEE

unigrams	F, AO, R, M, AX, S, IY
bigrams	F-AO, AO-R, R-M, M-AX, AX-S, S-IY
trigrams	F-AO-R, AO-R-M, R-M-AX, M-AX-S, AX-S-IY
…	…
7-grams	F-AO-R-M-AX-S-IY

Table 2. Possible n-grams extracted from an instance of PHARMACY fed through a phonetic recognition system.

The **recognition task** (A) provides the mapping from audio to a string of acoustic units using an ASR engine. Different pronunciation lexicons and language models can be used in the ASR engine to produce recognition results with different levels of basic acoustic unit. The details of the this task are described in section 2.1.

The **feature extraction task** (B) (summarized in section 2.2) uses the acoustic units to produce features that may be useful for the task. The features may be defined over the acoustic units themselves, or in the case of sub-word or word units, the acoustic units may be broken into their phonetic constituents before feature extractions.

The **IR scoring task** (C) consists of two options: vector space model (VSM) based scoring or language model based scoring, which are used to score the likelihood between a test query and each training document. The document with the highest score will be retrieved. These two schemes are described in details in section 2.3.

2.1. Recognition

The recognition task maps the audio into a sequence of acoustic units. In this paper, the same acoustic model is used regardless of the acoustic unit chosen. By pairing it with different pronunciation lexicons and language models, recognition results are obtained at different levels of basic acoustic units: phonetic recognition, multiphone recognition [8], and word recognition.

The parameters of this recognition component are summarized in section 3. As the size of the acoustic units is increased from phones to multi-phones to words, we expect the effective phonetic error rate decreases. Unfortunately, this comes at the cost of larger and more complex models. Also, the errors that remain are difficult to correct. For example, when "PHARMACIES" is misrecognized as "MACY'S," no subsequent processing can correct the error. Conversely, decreasing the size of the acoustic units will increase the effective phonetic error rate. But, the IR engine may have a good chance to recover from some errors as long as enough of the phones are correctly recognized.

2.2. Feature Extraction

The output of the recognition task will be a sequence of phones, multi-phones, or words. Features can be defined on this sequence, or the acoustic units can be first mapped into an equivalent phonetic string. Both approaches are tested in this paper.

The set of possible n-gram features on the recognition output is essentially limitless. For example, Table 2 enumerates some of the 28 possible n-gram units extracted from a single utterance. The entire training set contains millions of such features. We explore several different rules designed to select an appropriate subset of these n-gram features from the training data.

2.2.1. Bigram Unit Features

Bigram units are useful in our experiments for several reasons. First, the complete set of bigrams is not large: only about 1200 bigrams exist in the training data. Second, the bigram contains more sequencing information than the unigram features, which helps to reduce the effective homophones introduced when we ignore feature order. Third, when compared to longer units, they are more robust to recognition errors. For example, perturbing a single phone will change two bigram units in an utterance, but the same error will change three trigram units.

2.2.2. MMI N-Gram Unit Features

For units where a sufficient amount of training data is available, we compute the mutual information between the existence of that unit in a training example and the word labels. We use $I(u) = \{0, 1\}$ to indicate the presence or absence of a sub-word unit u. The mutual information between a unit u and all the words $\mathbf{W}$ in the training data is given by:

$$MI(u, W) = \sum_{I(u)} \sum_{w \in W} P(I(u), w) \log \frac{P(I(u), w)}{P(I(u))P(w)}. \qquad (1)$$

$P(I(u), w)$, $P(I(u))$ and $P(w)$ can all be estimated from counting procedure in the training data. Then we can rank all the sub-word units in training data based on the mutual information measure, and select only the highest ranked units.

2.3. IR Scoring

The goal of IR scoring is to quickly find the training document that most closely matches the testing query. The two scoring schemes, vector space model based IR and language model based IR, are discussed separately.

2.3.1. Vector Space Model Based Scoring

The vector space model (VSM) lives in a high dimensional feature space, where each dimension corresponds to one of the acoustic unit features discussed in section 2.2. In this section, to be consistent with the terminology used in IR, the acoustic unit feature will be called a "term" and the listing will be called a "document."

The essence of training the VSM is to construct a document vector for each document in the training data. This vector consists of weights learned or calculated from the training data. In our work, each training document represents a pool of examples that share the same listing. Each test example is interpreted as a query, composed of terms, which is also used to make a query vector.

The similarity between testing query q (with query vector $\mathbf{v}_q$ with elements v_{qk}) and training document d (with document vector $\mathbf{v}_d$ with elements v_{dk}) is given by their cosine similarity, a normalized inner product of the corresponding vectors.

$$\cos(\mathbf{v}_q, \mathbf{v}_d) = \frac{\sum_k v_{qk} v_{dk}}{|\mathbf{v}_q||\mathbf{v}_d|} \qquad (2)$$

TF-IDF

One simple method of computing the document vectors directly from the training examples is to use the popular TF-IDF formula from the information retrieval field. This weighting can be computed directly from counting examples in the training data as follows:

$$v_{jk} = \frac{f_{jk}}{m_j} \cdot (1 + \log_2(\frac{n}{n_k})), \quad j = q, d. \qquad (3)$$

In (3), $\frac{f_{jk}}{m_j}$ is the term frequency (TF), where f_{jk} is the number of times term k appears in query or document j and m_j is the maximum frequency of any term in the same query or document. $1 + \log_2(\frac{n}{n_k})$ is the inverse document frequency (IDF), where n_k is the number of training queries that contain term k and n is the total number of training queries.

A $N \times K$ term-document matrix is then created with the TF-IDF weighted training document vectors as its parameters. The rows represent the N terms and the columns the K training documents. The transpose of the term-document matrix is the routing matrix R

with its row $\mathbf{r}_i$ as the document vector. A query q is routed to the document $\hat{i}$ with the highest cosine similarity score

$$\text{document } \hat{i} = \arg\max_i \cos(\mathbf{v}_q, \mathbf{r}_i) \tag{4}$$

Discriminative Training

The routing matrix could also be discriminatively trained based on minimum classification error criterion using procedures similar to [9]. The discriminant function for document j and observed query vector $\mathbf{x}$ is defined as the dot product of the model vector and query vector

$$g_j(\mathbf{x}, R) = \mathbf{r}_j \cdot \mathbf{x} = \sum_k r_k x_k \tag{5}$$

Given the correct target document for $\mathbf{x}$ is c, the misclassification function is defined as

$$d_c(\mathbf{x}, R) = -g_c(\mathbf{x}, R) + \left[\frac{1}{K-1} \sum_{i \neq c, 1 \leq i \leq K} g_i(\mathbf{x}, R)^{\eta} \right]^{\frac{1}{\eta}} \tag{6}$$

Then the class loss function with L_2 regularization is

$$l_c(\mathbf{x}, R) = \frac{1}{1 + \exp^{-\gamma d_c + \theta}} + \lambda \sum_i \|\mathbf{r}_i\|^2. \tag{7}$$

The L_2 regularization is used to prevent over-fitting the training data and λ is set to be 100 in our experiments. All the other parameters in (6) and (7) are set the same as in [9]. The batch gradient descent algorithm with RProp [10] update is then used to search the optimum weights in the routing matrix.

MCE training is useful to demonstrate the best possible parameters for a given set of terms. It is unlikely that any simple choice of weighting function like TF-IDF will surpass the accuracy of an MCE trained system.

2.3.2. *Language Model Based Scoring*

A language model defines a probability distribution over sequences of symbols. Our language model based IR trains a language model for each document, and then the scoring is based on the probability of a training document d given a testing query q. The target correct document $\hat{d}$ for the query q can then be obtained:

$$\text{document } \hat{d} = \arg\max_d P(d|q) = \arg\max_d P(q|d)P(d). \tag{8}$$

In (8), $P(d)$ can be estimated by dividing the number of training queries in document d by the number of all training queries. Assuming the pronunciation of query q is $p_1, p_2, \cdots, p_m$, $P(q|d)$ can then be modeled by a n-gram language model:

$$P(q|d) = \prod_i P(p_i | p_{i-n+1}, \cdots, p_{i-1}; d), \tag{9}$$

where $P(p_i|p_{i-n+1}, \cdots, p_{i-1}; d)$ can be estimated by counting procedure. It is possible that a many n-grams are rarely seen or unseen in the training data, in which cases the counting would not give a reasonable estimate of the probability. In our experiments, Witten-Bell [11] smoothing scheme was used to calculate the discounted probability, which is able to smooth the probability of seen n-grams and assign some probability for the unseen n-grams.

3. EXPERIMENTS

The system is evaluated on a subset of the Windows Live Search for Mobile voice search task [12]. This data was divided into training(505994 utterances) and test(3621 utterances) partitions, and there are no out-of-vocabulary words in the test set. Each utterance in the data is a single voice query of business listing information. Experiments in this paper are restricted to using the "Top 1000" subset of the complete data. This data consists of those utterances corresponding to the one thousand most popular queries in the training data. By running experiments on this subset, we are able to reduce the complexity of our model while still training on a significant amount of data. Because these top 1000 queries contain some homophones, we consider only 981 unique listings.

h

Recognition	Type	Features(model)	Accuracy
Phonetic Recognizer + IR	Language Model (LM) Scoring	1gram-LM	65.9%
		2gram-LM	77.9%
		3gram-LM	79.0%
		4gram-LM	79.5%
		5gram-LM	79.2%
	VSM Scoring	bigrams + TF-IDF	71.3%
		bigrams + MCE-DT	82.4%
		MMI ngrams + TF-IDF	71.6%
		MMI ngrams + MCE-DT	83.3%

Table 3. Results from proposed system using phonetic recognizer for pronunciation recognition.

The same acoustic model used for all experiements in this paper. It contains 27760 diagonal Gaussian components that are shared among 863 states, which are in turn shared by 2612 hidden Markov models (HMM). These HMM model the context-dependent versions of 40 phone units, as well as 2 silence models and 3 garbage models.

This acoustic model can be paired with different pronunciation lexicons and language models to produce recognition results at different levels of pronunciation constraints, namely: phonetic recognition, multi-phone recognition, word recognition. In all these recognition systems, trigram language model is used. There are about 46 phone, 4.6k multi-phone and 65k word 1-grams.

3.1. Phonetic Recognition

Since the phonetic recognition is the most lightweight recognition task, it is the most attractive first stage of the system. Table 3 summarizes several experiments that use units drawn from the phonetic recognition system. We tested two styles of scoring the phonetic strings: a language model, and a vector space model .

For the first experiment, we used the language model based scoring to test the accuracy potentials when using phonetic recognition, where the language models were made by use of CMU SLM Toolkit [13]. We tested language models with order from 1gram to 5gram. The high order ngram is sparser than expected. For instance, the 4-grams in the training queries of the word "PHARMACY" only covers around 0.025% of all possible 4-grams. From the results shown in Table 3, 4gram LM acts best which proves that high order ngrams are useful for the task. The results of accuracy around 80% verifies that the units drawn from phonetic recognition contain certain useful information for our task.

We then did some experiments using vector space model based scoring to further test our proposed IR system. For the first two experiments, we used the complete set of bigram features for the IR system. TF-IDF feature weighting of the routing matrix (bigrams+TF-IDF) yielded only 71.3% accuracy. With the MCE discriminative training of the routing matrix (bigrams+MCE-DT), the accuracy can be greatly improved to be 82.4%. For the third experiment, the 50000 highest ranked MMI units are selected as features for the IR system, TF-IDF feature weighting of the routing matrix (MMI ngrams+TF-IDF) still yielded only 71.6% accuracy. Furthermore, MCE discriminative training is used on the routing matrix (MMI ngrams+MCE-DT). Even then, the accuracy increases to only 83.3%.

These results indicate that, although phonetic recognition may contain enough information to produce a good n-best list to be further processed [5], it is too noisy for a first-pass recognition.

Recognition	Type	Features(model)	Accuracy
Multi-phone Recognizer + IR	Language Model Scoring	2gram	90.1%
		3gram	90.5%
		4gram	90.4%
	VSM Scoring	multi-phone + TF-IDF	88.2%
		multi-phone + MCE DT	91.0%

Table 4. Results from proposed system using phonetic recognizer for pronunciation recognition.

3.2. Multi-phone Recognition

Since phonetic recognition seems to be too noisy, we add some constraints to the recognition system and multi-phone units are chosen as the acoustic units in our proposed system. These multi-phone units are sub-word units that usually contains 3 to 4 phones. Table 4 demonstrates the behavior of using the multi-phone units as IR features. To compare with the phonetic recognizer, firstly we mapped the multi-phone units into their corresponding phones and used the language model based scoring. The over 90% accuracy indicates that the multi-phone recognition is much less noisy than the phone recognition. So we tried to directly use those multi-phone units as the acoustic unit features of our proposed vector space model based IR system. The 88.2% of accuracy by using the TF-IDF weighting can be improved to 91.0% with the MCE discriminative training.

3.3. Adding Constraints to the Recognizer

In this section, we compare three levels of constraints in the recognition system. In addition to the phonetic and multi-phone recognition results already discussed, we also constrain the recognizer to output only valid word units according to a fixed dictionary and language model. In all of these systems, the longer units were mapped into their corresponding phones before extracting features for the IR system. Table 5 shows the results using these three different decoding units in the proposed system. Three different feature and model setup are compared, including VSM scoring with bigram features combining either TF-IDF weighting or MCE DT training of the routing matrix, and LM scoring with best choice of ngram order.

Using word recognition, there is little benefit for using MCE-DT compared to TF-IDF in VSM based scoring, and little difference is observed between VSM and LM based scoring. This indicates that the errors from word recognizer are difficult to correct.

The best accuracy is achieved using multi-phone recognition. We believe the multi-phone units strike a good compromise between performing phone and word recognition.

It is useful to compare two different HMM baselines in this task: word recognition, and utterance classification. In the conventional word based HMM based recognition, the full vocabulary and language model is used, and achieves an accuracy of 86.6%. But, because the IR system is essentially performing classification, we should also compare to an HMM based classifier. The utterance classification system uses the same acoustic model and pronunciation dictionary as the word recognition system, but its output is restricted to be one of the 981 unique listings. Because utterance classification is easier than word recognition, this system reaches 90.3% accuracy.

In all our experiments, the multi-phone recognizer leads to the best system which has 12% fewer errors than the comparable HMM classifier, and even using an inexpensive IR weighting scheme (TF-IDF) yields a 3% relative error rate reduction.

4. CONCLUSIONS

We proposed a new speech recognition system. The integration of information retrieval techniques in this system provides a flexible pronunciation modeling for the ASR. The proposed system is able to outperform a traditional HMM based ASR system.

	Recognition Unit		
Features(model)	Phone	Multi-phone	Word
bigram+TF-IDF(VSM)	71.3%	90.6%	87.6%
bigram+MCE DT(VSM)	82.4%	91.4%	87.9%
best ngram(LM)	79.5%	90.5%	88.2%

Table 5. Results of recognition accuracy from proposed system using three different units for recognition.

5. ACKNOWLEDGEMENTS

The authors wish to thank our colleagues Geoffrey Zweig for helping setup the multi-phone unit decoding, Xiao Li and Wei Wu for helping setup the language model based scoring and Chin-Hui Lee for many useful discussions in this project.

6. REFERENCES

[1] H-K.J. Kuo and Y. Gao, “Maximum engropy direct models for speech recognition,” *in Proc. of ASRU*, 2003.

[2] A. Gunawardana, M. Mahajan, A. Acero, and J. Platt, “Hidden conditional random fields for phone classification,” *in Proc. of Interspeech*, 2005.

[3] G. Heigold, G. Zweig, X. Li, and P. Nguyen, “A flat direct model for speech recognition,” *in Proc. of ICASSP*, 2009.

[4] Y.-Y. Wang and A. Acero, “Maximum entropy model parameterization with tf-idf weighted vector space model,” *in Proc. of ASRU*, 2007.

[5] A. Moreno-Daniel, B.-H. Juang, and J. Wilpon, “A scalable method for voice search to nationwide business listings,” *in Proc. of ICASSP*, 2009.

[6] O. Siohan and M. Bacchiani, “Fast vocabulary-independent audio search using path-based graph indexing,” *Interspeech*, pp. 53–56, 2005.

[7] K. Ng and V. W. Zue, “Subword-based approaches for spoken document retrieval,” *Speech Communication*, vol. 32, pp. 157–186, 2000.

[8] G. Zweig and P. Nguyen, “Maximum mutual information multi-phone units in direct modeling,” *in Proc. of Interspeech*, 2009.

[9] H-K.J. Kuo and C.-H. Lee, “Discriminative training in natural language call routing,” *in Proc. of ICSLP*, 2000.

[10] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: The RPROP algorithm,” *in IEEE Int. Conf. on Neural Networks*, 1993.

[11] I. Witten and T. Bell, “The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression,” *IEEE Transactions on Information Theory*, vol. 37, no. 4, 1991.

[12] A. Acero, N. Bernstein, R. Chambers, Y.C. Ju, X. Li, J. Odell, P. Nguyen, O. Scholz, and G. Zweig, “Live search for mobile: Web services by voice on the cellphone,” *in Proc. of ICASSP*, 2008.

[13] P. Clarkson and R. Rosenfeld, “Statistical language modeling using the cmu-cambridge toolkit,” *in Proc. of EUROSPEECH*, 1997.