Maximum Mutual Information Multi-phone Units in Direct Modeling

Geoffrey Zweig, Patrick Nguyen

Microsoft Research, Redmond, WA

{gzweig,panguyen}@microsoft.com

Abstract

This paper introduces a class of discriminative features for use in maximum entropy speech recognition models. The features we propose are acoustic detectors for discriminatively determined multi-phone units. The multi-phone units are found by computing the mutual information between the phonetic subsequences that occur in the training lexicon, and the word labels. This quantity is a function of an error model governing our ability to detect phone sequences accurately (an otherwise informative sequence which cannot be reliably detected is not so useful). We show how to compute this mutual information quantity under a class of error models efficiently, in one pass over the data, for all phonetic sub-sequences in the training data. After this computation, detectors are created for a subset of highly informative units. We then define two novel classes of features based on these units: associative and transductive. Incorporating these features in a maximum entropy based direct model for Voice-Search outperforms the baseline by 24% in sentence error rate.

Index Terms: speech recognition, direct model, maximum mutual information, features

1. Introduction

In recent years, there has been significant interest in direct modeling for speech recognition, an approach which holds out the promise of learning mappings directly from acoustic features to word sequences. Compared to traditional generative models, the benefits that these approaches offer include an inherently discriminative training process, and the ability to reason and compute in terms of numerous and possibly redundant features. Two previously proposed approaches in this vein are Maximum-Entropy Markov Models (MEMMs) [1] and Conditional Random Fields (CRFs) [2, 3]. In the first approach, a maxiumum entropy model is used to determine the probabilities of state transitions given acoustic features, in an HMM that still encodes conventional sequencing constraints from the lexicon and decision tree. In the CRF approach, the total path probability is factored as the product of state-state transition functions, and state-feature observation functions. Similar to the MEMMs approach, it uses conventional sequencing constraints on the allowable state sequences. Both these techniques may be viewed as hybrid HMM/Direct-Model approaches, in that they inherit the chain-like state sequencing of traditional HMMs, while using more advanced models to compute the probability of different sequences.

We recently proposed a more radical direct modeling approach in [4]. In this approach, a set of "consistency features" is defined between a linguistic hypothesis and the underlying acoustics. The posterior probabilty of the linguistic hypothesis is then given by a maximum entropy model on the features. More precisely, if there is a set of linguistic hypotheses N for an utterance with acoustics x, then the probability of a specific hypothesis $h \in N$ is given by

$$P(h|X) \;=\; \frac{\exp(\sum_i \lambda_i \kappa_i(x,h))}{\sum_{n\in N} \exp(\sum_i \lambda_i \kappa_i(x,h))}. \qquad (1)$$

For an important class of features, the feature functions may be decomposed as $\kappa_i(x,h) = \psi_i(x)\phi_i(h)$. In this formulation, $\phi_i(h)$ is a yes/no linguistic feature of the hypothesis, for example that it ends in an "s". In contrast, $\psi_i(x)$ is a yes/no feature of the acoustics themselves, for example that a sibilant has been detected near the end of the utterance. The conjunction between a lingusitic and an acoustic feature is essentially one measure of the consistency between the hypothesis and the underlying acoustics, for example that the hypothesis ends in "s" and a sibilant near the end was detected. This approach can be generalized by using real-valued rather than binary features, and by defining features that are non-zero when an *inconsistency* is detected. For convenience, we will refer to the models we discuss, using both factored and unfactored features, as *Consistency Feature Direct Models* or CFDMs.

CFDMs differ from CRFs and MEMMs in that they are not defined in terms of conventional state sequences, but only in terms of consistency features. In this sense they are more similar to maximum entropy based trigger language models [5]. It is also important to note that while maximum entropy systems are often associated with n-way classification tasks, CFDMs in fact generalize to completely unseen hypotheses: as long as the consistency features can be computed, the approach can be used. Like all maximum entropy based approaches, the key issue becomes exactly what features are used. In [4], letter n-gram linguistic features were used, along with HMM-based hypothesis posteriors and word-level template distances. In this paper, we break up the problem with the following Markov chain:

$$x \rightarrow \{u\} \rightarrow \{w\} \rightarrow h, \qquad (2)$$

where x is the audio, u are multi-phone units, w are words, and h is a concept (e.g. a business identity). In this paper, we attempt to maximize the transfer of information from x to w by carefully selecting u; that is, we want acoustic features $\{\psi_i(x)\}$ that are known to have a high degree of mutual information with word labels. In this paper, we show how to compute this quantity for multi-phone sequences in the lexicon, and create acoustic detectors for highly informative units. Experimental results are reported on a voice-search task in which we must recognize queries for business names, such as "Bed Bath and Beyond" or "Kragen Auto Parts."

The remainder of this paper is organized as follows. In Section 2, we present the computation of mutual information between multi-phone units and word labels. Section 3 defines the consistency features we use in our models. Section 4 describes the search strategy we have used in decoding; then in Section 5, we present our experimental setup and results. Section 6 offers concluding remarks.

2. MMI Multi-phones

In this section, we outline the process for identifying informative multi-phone units. The input to this process is a dictionary that indicates the phonetic spelling for each word, along with the unigram counts for each word. We will assume at first that each word has one pronunciation. The output will be the mutual information between each phonetic sequence in the lexicon, and the word labels. The phonetic sequences are arbitrary sub-word units (e.g. "aa k iy" from akimoto) that may span anything from a single phone to an entire word.

2.1. The Errorless Case

We use $u_j = \{0,1\}$ to denote the presence or absence of a multi-phone unit. The mutual information between a unit u_j and the words is then given by:

$$\begin{aligned} MI(u_j;w) &= \sum_{a=\{0,1\}} \sum_{w=w_k} P(u_j=a,w) \log \frac{P(u_j=a,w)}{P(u_j=a)P(w)} \\ &= \sum_w P(w)p(u_j=1|w) \log \frac{P(u_j=1|w)}{P(u_j=1)} \\ &\quad + \sum_w P(w)P(u_j=0|w) \log \frac{P(u_j=0|w)}{P(u_j=0)}. \end{aligned}$$

If we then break the set of words up into those in which u_j is present (w^+) and those in which it is not present (w^-), we may take advantage of the fact that $P(u_j=1|w^-)=0$ and $P(u_j=0|w^+)=0$ to simplify:

$$\begin{aligned} &MI(u_j;w) \\ &\quad = -\sum_{w^+} P(w^+) \log P(u_j=1) - \sum_{w^-} P(w^-) \log P(u_j=0) \\ &\quad = -\log P(u_j=1) \sum_{w^+} P(w^+) - \log P(u_j=0) \sum_{w^-} P(w^-) \\ &\quad = -\log(\sum_{w^+} P(w^+)) \sum_{w^+} P(w^+) - \log(\sum_{w^-} P(w^-)) \sum_{w^-} P(w^-). \end{aligned}$$

In one pass over the data, we can compute $\sum_{w^+} P(w^+)$ and $\sum_{w^-} P(w^-) = 1 - \sum_{w^+} P(w^+)$ for any unit u_j, and in fact by examining the words one-by-one and updating counts for all the phoneme sub-sequences present, we can accumulate the sum for *every* unit u.

2.2. The Effect of Errors

In reality, our ability to detect unit presence is imperfect. An otherwise highly informative unit that cannot reliably be detected is in fact not so useful. From Eq (2), units which are good for $u \rightarrow w$ might not be suitable because $x \rightarrow u$ is weak. Considering errors, four outcomes are possible when we attempt to detect whether a unit is present: a correct accept, a false reject, a false accept, and a correct reject. Taking this into account:

$$\begin{aligned} &MI(u_j,w) \\ &\quad = \sum_w P(w)P(u_j=1|w) \log \frac{P(u_j=1|w)}{P(u_j=1)} \\ &\quad + \sum_w P(w)P(u_j=0|w) \log \frac{P(u_j=0|w)}{P(u_j=0)} \\ &\quad = \sum_{w^+} P(w^+)P(u_j=1|w^+) \log \frac{P(u_j=1|w^+)}{P(u_j=1)} \text{correct accept} \\ &\quad + \sum_{w^+} P(w^+)P(u_j=0|w^+) \log \frac{P(u_j=0|w^+)}{P(u_j=0)} \text{false reject} \\ &\quad + \sum_{w^-} P(w^-)P(u_j=1|w^-) \log \frac{P(u_j=1|w^-)}{P(u_j=1)} \text{false accept} \\ &\quad + \sum_{w^-} P(w^-)P(u_j=0|w^-) \log \frac{P(u_j=0|w^-)}{P(u_j=0)} \text{correct reject} \end{aligned}$$

Unit	$MI(\mu_j;w)$
ax_n	0.026 bits
k_ae_l_ax_f_ao_r_n_y_ax	0.023
ae_l_ax_f_ao_r_n_y_ax	0.022
k_ae_l_ax_f_ao_r_n_y	0.022
l_ax_f_ao_r_n_y_ax	0.022

Table 1: The most informative multi-phone units. An "_" is used between the phones belonging to a single unit.

Unit	$MI(\mu_j;w)$
ax_n	0.026 bits
k_ae_l_ax_f_ao_r_n_y_ax	0.023
ax_r	0.021
s_t	0.018
ao_r	0.017

Table 2: The most informative multi-phone units after unit selection.

This can be computed efficiently for each candidate multi-phone unit in the data in two steps. In the first, a single pass over the data is made, and we compute the same quantities that were used in the errorless case. In the second, each candidate unit u_j is examined, and the mutual information computed according to the above formula. We have used an error model in which the probability of a false accept is exponentially decreasing in the length of the units, and in which the probability of a false reject is constant. The necessary quantities are readily available:

- $\sum_{w^+} P(w^+)$ and $\sum_{w^-} P(w^-)$ are computed in the initial pass over the data for each u_j as in the errorless case.
- $P(u_j=1|w^-) = ae^{-bl}$ where l is the length of the unit in phones, and a and b are constants.
- $P(u_j=0|w^-) = 1 - P(u_j=1|w^-)$
- $P(u_j=0|w^+) = c$, a constant
- $P(u_j=1|w^+) = 1 - P(u_j=0|w^+)$
- $P(u_j=1) = P(u_j=1|w^+)\sum_{w^+} P(w^+) + P(u_j=1|w^-)\sum_{w^-} P(w^-)$
- $P(u_j=0) = P(u_j=0|w^+)\sum_{w^+} P(w^+) + P(u_j=0|w^-)\sum_{w^-} P(w^-)$

Note that in the last two bullets we have taken advantage of the fact that $P(u_j=\{0,1\}|w^{\{+,-\}})$ is only a function of u_j to move this factor outside the summations. In our experiments, we have used $a=1$,b=1,$c=0.5$.

The approach outlined above is straightforward to implement with a single pronunciation for each word. When multiple pronunciations are present, the quantities that must be computed do not factor so neatly. However, by using the mutual information between words and pronunciation variants, one obtains a useful surrogate. Alternatively, one may determine the unit set simply by using the most common pronunciations. In the experiments below, we used the first approach.

2.3. Unit Selection

Table 1 shows the five most informative multi-phone units. As can be seen, many of these units derive from the word "California" (our training data included city-state-zip requests), and from the point of view of building detectors, it would be inefficient to use such redundant units. (The redundancy stems from the fact that while each unit has a large amount of mutual information with the words, the conditional mutual information

Word	Unit Breakdown
Academia	ae_k_ax d_iy m_iy ax
Academic	ae_k_ax d_eh m_ih_k
Academics	ae_k_ax d_eh m_ih_k s
Academies	ax_k_ae_d_ax_m_iy z
Academy	ax_k_ae_d_ax_m_iy

Table 3: Segmentation of several words into multi-phone units.

Word	Unit Breakdown
Pizza	p_iy_t_s_ax
Wal-Mart	w_aa_l_m_aa_r_t w_ao_l_m_aa_r_t
McDonald's	m_ih_k_d_aa_n_ax_l_d_z
Best Buy	b_eh_s_t b_ay
Starbucks	s_t_aa_r_b_ah_k_s

Table 4: Segmentation of the most common requests into multi-phone units.

of one unit given another is low.) To find a less redundant set of units, we proceed by decaring a set of candidate units - the top $N = 10000$ most informative - and then partitioning each dictionary word into the minimum number of candidate units. Any unit that is not selected is then thrown away. This results in a set of 5662 units. The most informative of the selected units are shown in Table 2. To get a sense of how the units are used, Table 3 shows the segmentation of several words. This illustrates how words of modest frequency are typically decomposed into syllable-like and single-phone units. Table 4 shows the segmentation of the most common business requests, and illustrates the fact that very common words typically result in whole-word units.

3. Definition of Features

Our features are based on Eq (2), and make reference to a lattice of decoded multi-phone units u. This lattice is created by decoding an utterance with multi-phone rather than word-level units. From this lattice and statistics derived from it, we extract two major kinds of features: associative, and transductive. Associative features provide indicators of what words might be expected on the basis of the units that are present, irrespective of ordering constraints. Transductive features then incorporate ordering information. Additionally, we use a set of background features. All these features make use of quantities defined by a simple model for determining the probability of a hypothesis:

$$p(h|x) = \sum_{u,w} p(u|x)p(w|u)p(h|w), \tag{3}$$

with:

$$p(h|w) = \frac{p(h)}{p(w)}\delta(w \in h), \tag{4}$$

and $\delta(\cdot)$ is the indicator function. In contrast with a standard HMM, $p(w|u)$ is not determined by a pronunciation lexicon: rather, it is a bag model. $p(w|u)$ is the ML estimate derived from decoding a held out portion of the training set, and counting how often u and w co-occur *at the utterance level*. $P(w)$ and $P(h)$ are similarly computed from this held out data. We use $\gamma(\cdot|x)$ to denote a posterior count. The next sections describe the feature types in detail.

Name	$\psi(x)$	$\phi(h)$	N
letters	1	$\delta(ngram \in h)$	10^6
prior	1	$\delta(h = T)$	100
S	$\gamma(u \mid x)\delta(u$ is terminal$)$	$\delta(h$ ends in s$)$	5k

Table 5: Background features.

Name	$\psi(x)$	$\phi(h)$	N
uw	$\gamma(u \mid x)p(w \mid u)$	$\delta(w \in h)$	10^6
word	$\sum_u \gamma(u \mid x)p(w \mid u)$	$\delta(w \in h)$	30k
hyp	$\kappa(x,h) = \log p(h \mid x)$		1

Table 6: Associative features. The first line defines one ψ feature for each u, w combination; the second one for each w. The third is defined at the utterance-level.

3.1. Background Features

As background features present in all our experiments, we introduce language model factors $\phi(h)$ that indicate the presence of letter 6-grams in h, similar to [4, 6]. Additionally, we use a language model feature to indicate the presence of each of the 100 most frequent requests, essentially to adjust their priors in a discriminative way. Since our baseline system often confuses plural and singular instances of entities, we also define an "s-at-the-end" feature, **S**. These are summarized in Table 5.

3.2. Associative Features

The associative features we use are illustrated in Table 6. The first of these measures the consistency between the presence of a word in the hypothesis, and its expected count as evidenced by the presence of a particular unit u in the unit lattice. The second feature measures the consistency between the presence of a word in the hypothesis, and its expected count marginalized over the contribution of *every* unit in the lattice. Finally, we introduce a feature indicating the posterior probability of a hypothesis as determined by Eq (3).

Reference	Recognized as
Marriott Courtyard	Courtyard Marriott
Harley-Davidson	Motorcycles
Borders	Borders Books
Gentlemen's Club	Adult Entertainment

Table 7: Recognition errors with associative features.

Our associative features are computed over an entire utterance and allow global relationships to be modeled. For example, since the unit "b_ay" is present in utterances of *Best Buy*, it will trigger for *Best Buy*, *Buy-Rite* and *Seattle's Best.* In early stages of development, we noticed that this property causes some interesting mis-recognitions, as shown in Table 7. For instance, many of our users in the training set asked for *Harley-Davidson Motorcycles.* During test, units representing the sound "Harley" (e.g. hh_aa_r) provided acoustic evidence for both *Harley-Davidson*, and *Motorcycles*, but since the *Motorcycles* language model was stronger, the latter was chosen. These errors underline the associative, rather than transductive nature of the approach. They disappeared when we introduced the next set of features.

3.3. Transductive Features

So far we have not introduced any features to measure the consistency between the order in which units are detected, and the

Name	$\kappa(x, h)$	N
ins	$\gamma(u\|x) \cdot \delta(u$ is an insertion in h)	5k
del	$\delta(u$ is a deletion from h)	5k
sub	$\gamma(u\|x) \cdot \delta(u$ is substituted in h)	5k
corr	$\gamma(u\|x) \cdot \delta(u$ is matched in h)	5k

Table 8: Transductive (Levenshtein) features are extracted from Levenshtein alignment.

order in which they are expected based on a hypothesis. To do this, we associate h with a pronunciation graph of how its surface sequence of words may be turned into units. We then compute the Levenshtein alignment between this graph and the unit lattice of x, and add features that indicate the number of edits that were necessary. The larger the number of edits, the less the consistency between the acoustic-based unit graph and the linguistic hypothesis. These features are all un-factored, and are illustrated in Table 8. They are implemented with rational operations [7].

4. Search Strategy

Given the ability to detect multi-phone units in an utterance, one must still be able to use these to determine the likeliest words. For any particular hypothesis, the consistency features mentioned in Section 1 can be computed and the maximum entropy score found. Therefore, the problem boils down to finding an appropriate set of hypotheses to score.

To that end, we implemented Eq (3) as a first pass, using Eq (4) to estimate $P(h|w)$. First, we compute the word posterior counts from the unit lattice, using $\gamma(w|x) = \sum_u p(w|u)\gamma(u|x)$. Then, we retrieve all relevant hypotheses by taking Eq (4) into account. Finally, hypotheses with low $p(h|x)$ score are pruned. In practice, we find that we can reduce the search space by 90% without incurring search errors.

5. Experiments

We applied CFDMs to recorded real-world interactions from the Windows Live for Mobile application [8]. This application allows users to request local businesses by voice, from their mobile phones. Key to our acquisition of training data, once a query is spoken, a list of alternatives is presented for user validation. Speech comes in various challenging conditions, including outside noise, music, side-speech, sloppy pronunciation, and different acquisition channels.

For the purpose of this paper, we set aside 3619 human-transcribed interactions for evaluation. For simplicity, we made sure that these queries were in the 1000 most popular, which covers around 41% of total query mass. For training, we availed ourselves of roughly 3M spoken queries – 2500h of speech. The transcriptions used for these queries were the items the users selected from the N-best lists. We estimate that this form of supervision is about 90% accurate. Furthermore, we divided the 3M training set into two parts: a hyper-parameter training set, and a model training set. We reserved the former set to build feature generators, while the log-linear model weights were optimized on the model training set. Since the real intention of the user is unobservable *ex post*, we measured sentence error (SER) instead, which in general is an overstimate of semantic error.

The baseline system is a conventional HMM system, using utterance-level mean normalized MFCCs and clustered cross-word triphones. It has 10,900 context dependent states, and altogether 260k Gaussians. This baseline was trained on all 2500 hours of speech and produces an error rate of 13.4%. The baseline was restricted to the vocabulary of the 1000 most frequent queries, and the LM was optimized for that data, so this is the fair comparison baseline achievable by the HMM approach.

To generate unit lattices, we used a trigram language model on units. Using Eq (3), we observed an error rate of 31% – higher than the baseline because it uses a weaker bag-of-words model, no word-to-unit lexicon, and a weaker unit language model.

We summarize results with the proposed features in Table 9. Unlike [4], we did not include HMM features, so we were not guaranteed its error rate as an upper bound. Rather, we would have obtained 31% had we used solely hyp features. Business and n-gram prior features (Table 5) were always included. Note that uw features are insufficient to represent the mixture model of Eq (3). However, we see no improvement by adding the hyp feature: linearization was a good approximation.

System	SER
HMM	13.4%
uw	12.5%
+hyp	12.5%
+word	12.4%
+terminal **S**	12.1%
+Levenshtein	10.2%

Table 9: Sentence error rate (SER) with various feature sets.

6. Conclusions

In this paper, we have proposed a class of direct modeling features which consist of multi-phone units that maximize the transfer of information between audio and words. Further, we have shown how to extract these units efficiently. We have used these units in the CFDM framework as “detectors”. Based on the detection of these units, we have defined a set of associative features which exhibit a different class of errors than might be expected from HMMs. Further, we have used unit-to-word transduction features to incorporate pronunciation knowledge. These features are integrated in a log-linear model which is free to choose the best combination. Combining all features, we have outperformed the HMM baseline by 24%.

7. References

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

[2] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. of 18th International Conf. on Machine Learning*, 2001.

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

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

[5] R. Lau, R. Rosenfeld, and S. Roukos, “Trigger-based language models: A maximum entropy approach,” in *Proc. of ICASSP*, 1993.

[6] B. Roark, M. Saraclar, and M. Collins, “Discriminative n-gram language modeling,” in *Computer, Speech and Language*, 2006.

[7] C. Cortes, P. Haffner, and M. Mohri, “Rational kernels: Theory and algorithms,” *Journal of Machine Learning Research*, vol. 5, pp. 1035–1062, 2004.

[8] A. Acero et al., “Live search for mobile: Web services by voice on the cellphone,” in *Proc. of ICASSP*, 2007.