Speech Recognition with Flat Direct Models

Patrick Nguyen, *Member, IEEE,* Georg Heigold, and Geoffrey Zweig, *Senior Member, IEEE*

***Abstract*—This article describes a novel direct modeling approach for speech recognition. We propose a log-linear modeling framework based on using numerous features which each measure some form of consistency between the underlying speech and an entire sequence of hypothesized words. Since the model relates the entire audio signal to a complete hypothesis without necessarily positing any inherent structure, we term this a Flat Direct Model (FDM). In contrast to a conventional HMM approach, no Markov assumptions are used, and the model is not necessarily sequential. We demonstrate the use of features based on both template-matching distances, and the acoustic detection of multi-phone units which are selected so as to have maximal mutual information with respect to word labels. Further, we solve the key problem of how to define features which can generalize to unseen word sequences. In the proposed model, template-based features improve sentence error rate by 3% absolute over the baseline, while multi-phone based features improve by 2% absolute.**

***Index Terms*—speech recognition, direct model, maximum mutual information, log-linear model, features.**

I. Introduction

GENERATIVE Hidden Markov Models (HMMs) stand at the heart of all large vocabulary state-of-the-art speech recognition systems today. With current adaptation and discriminative training techniques, the approach is efficient, effective, generalizes well, and is eminently practical [1], [2], [3]. Nevertheless, several concerns - some new and some long-standing - indicate that it might be worthwhile to explore alternative methodologies. The first concern is the use of a frame-level Markov property. The simplicity afforded by this property is a key source of computational and mathematical tractability, but it is mainly an engineering expedient, rather than being desirable from a scientific viewpoint. Over the years, articulate criticism has been leveled at this property from speech science (*e.g.* [4]), and we have empirical evidence [5], [6] that long-span dependencies may be captured at the acoustic level and profitably fed back in the feature vectors.

The second reason for exploring alternative methods is that the generalization capabilities we see with HMMs come from the massive use of parameter tying. As more data become available, it is reasonable to question how far these models can be extended with a combination of whole-word models, ever larger decision trees, and Gaussian mixtures. In contrast, Nearest Neighbor (NN) classifiers, which grow proportionally with the size of the training data, can significantly outperform models with fixed and limited number of parameters when data abound. Thus, as more data have become available in recent years, there has been a resurgence of Dynamic Time Warping (DTW) techniques and coarse-to-fine approaches to speech recognition [7], [8].

Lastly, we are seeing commercial interest in Voice Search applications [9], [10], [11], [12], [13]. By voice search, in this paper, we mean the ability to look up business information such as phone numbers and addresses by voice. The voice search applications are distinct in their characteristics from more traditional large vocabulary continuous speech recognition (LVCSR) tasks, such as dictation or human to human phone conversations. In particular, the distribution over output strings is heavily weighted towards the most common requests. In a dictation setting, for instance, in a typical Wall Street Journal (WSJ) sentence, there are 17 words drawn from a quality-filtered 64k vocabulary. Notionally, the output space would be $64000^{17} = 2^{272}$. In such a setting, considering each hypothesis individually during search is computationally prohibitive. In the Voice Search application, however, we see a different behavior. Figure 1 shows that the empirical mass captured by events considered, drawing from a trigram language model, *without looking at any audio*. Note that we can capture a large share of the probability mass with just a few thousands of entries – and improving on those queries will help disproportionately to their number. Further, there are only about 30M businesses in the United States, and picking the correct one is easier than solving the ASR task. Therefore, in Voice Search, since speech comes in short phrases which are typically two or three words long, it is natural to question the judiciousness of treating the problem as a degenerate case of LVCSR.

In this paper, we define *Flat Direct Models* (FDMs) to address these concerns. These models have two key characteristics. First, they are *direct* in the sense defined by [14] in that they model the posterior distribution of the desired output (a sequence of words), conditioned on the input audio, rather than having a generative or joint model generating audio from a sequence of desired textual representation and flipping it backwards with Bayes rule during recognition. In other words, they are *conditional* models: the probability of an output is conditioned on given input audio. In practice, our FDMs are log-linear models of a feature vector defined as a function of the whole audio sequence and the output hypothesis.

Second, and most importantly, FDMs are *flat*. That is, the model is defined at the utterance level, and may or may not make reference to lower level structure, such as word or phoneme ordering. In our experiments, we use both structured and unstructured features. In the Voice Search application, an index may be associated to each of the 30M businesses in the United States, and the goal is to guess which is the right index. In practice, for operational purposes, the real intent of the user is unobservable, and we approximate it by a text transcription. Nevertheless, there no presumption of an inherent notion of

P. Nguyen and G. Zweig are with Microsoft Research, {panguyen,gzweig}@microsoft.com

G. Heigold is with RWTH Aachen University

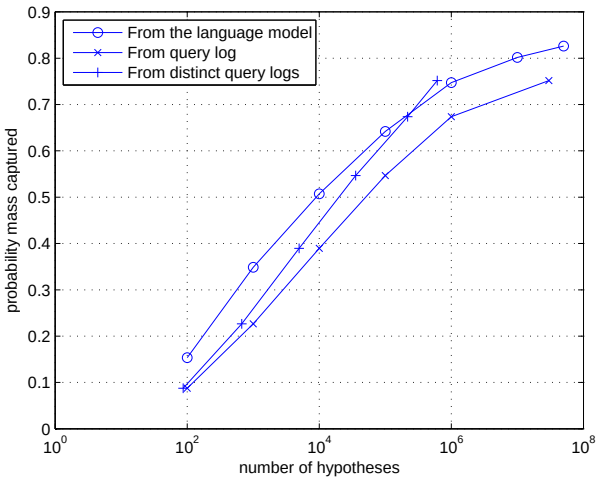

Fig. 1. In-grammar rates on the Voice Search task for different sizes of hypothesis space. For hypotheses generated by the language model, we drew the most likely sequences in order. For the query logs, we drew samples in chronological order.

words, causal order among them, or local contiguous statistical dependency required in the approach, much less that words need to be constructed from subword units such as triphones.

Despite this generality, FDMs have the nice property that by adding HMM-based features such as HMM acoustic and language model scores, the FDM is always guaranteed to perform no worse than the HMM system on average - this is an important feature which is not always shared by novel non-HMM approaches. In practice, this means that the model can make use of new embodiments of information without the need for painstakingly deriving an alternative architecture that is competitive with HMMs from scratch.

In contrast with HMMs, since the models do not construct words by concatenating subword units, they are a natural fit for embedding long-span phenomena. In fact, as outlined below, the order of words is considered only if convenient, and to the extent required. We can even consider completely global features - for instance, arbitrary duration distributions for specific words.

This paper makes several contributions. First, we define and describe the notion of Flat Direct Models. The model quality is determined by the features which are used, and our second contribution is to propose two main types of features: template-based, and detector-based. In the template-based approach, we show that we can successfully add template features to an HMM baseline. In the detector-based approach, we solve two key problems: what acoustic detectors to use, and how to define features that generalize to word sequences which are unseen in the training data. To solve the acoustic processing problem, we propose the use of features based on the detection of discriminatively determined multi-phone units. These units are determined by computing the mutual information between phonetic subsequences that occur in the lexicon, and the word labels. Based on this criterion, detectors are created for a subset of highly informative units. Then, to solve the feature generalization problem, we develop the notion of decomposable features that consist of the conjunction of a purely acoustic part and a purely linguistic part, and additionally the use of associative and transductive features.

The remainder of this paper describes the flat direct modeling framework in detail. We draw together material presented in [15], [16], and present new error models for use in the computation of MMI multi-phone units [16], as well as an efficient exact method for computing the mutual information between candidate phoneme sequences and words. The paper further presents new comparative results with these various techniques.

The remainder is organized as follows. Section II presents the functional form of the FDM. Section III describes the classes of features we use, and in particular how these solve the generalization problem wherein a *flat* model can nevertheless be trained with one set of words or listings, and then successfully used even with unseen words and listings. Section IV describes a new class of acoustic detectors - MMI Multi-phones - for use with our models. These units have been designed to provide a great deal of information about the words, while simultaneously being robust to expected acoustic modeling errors. Section V discusses potential limits with the approach. In Section VI, we present a suite of experiments evaluating our model and features, followed in Section VII by concluding remarks.

II. Model: functional form

The Flat Direct Model is implemented as a log-linear model. If we receive a sequence of audio X, and denote desired output (such as a the identity of a business) by h, our model is:

$$p(h|X) := \frac{\exp\left[\lambda^T f(h,X)\right]}{\sum_{h'} \exp\left[\lambda^T f(h',X)\right]}. \quad (1)$$

The weight vector λ defines the model parameters, and the feature vector $f(h,X)$ is a function of both the audio and the output. Unlike Maximum-Entropy Markov Models [14] or Hidden Conditional Random Fields [17], this model is *flat* and does not have the notion of the sequence: it is informed by sequential information only through the feature vector. The crucial question of what features to define will be explored in the next section.

In our experiments, the FDM is trained with a corpus of T labeled utterances $\{h_t, X_t\}_{t=1}^T$ with a regularized log-likelihood J [18]:

$$J := \sum_t \log p(h_t|X_t) - \sigma^{-2}||\lambda||^2/2. \quad (2)$$

The prior weight σ^{-2} ranged from 10^3 to 10^7. Its value was set by optimizing the error rate on a development set, searching among values which were integral powers of ten. Model weights were estimated using RProp [19], initialized with $\lambda = 0$, and run for exactly 40 iterations, which was sufficient to observe convergence in J and error rate on the development set.

III. Classes of features

As can be seen from Eq (1), it is possible to define arbitrary feature functions $f(h, X)$, for instance, the consistency of the pitch contour with the putative pronunciation of a business name. In this paper, however, we explore a restricted subset of possible features classes of which we now provide a taxonomy. The features of Sections III-A, III-B, III-C first appeared in [15], while those of Section III-D first appeared in [16]. Before proceeding, we note that the features $f(h, X)$ may make use of different representations of the audio X. For example, MFCCs may be used, or a sequence of detected subword units such as phones may be substituted for the raw cepstra. In either case, the representation of the audio is not in itself a feature. As we shall see, the acoustic information must be related to the linguistic hypothesis h before becoming a feature $f(h, X)$ in our model.

A. HMM Baseline features

As noted in the introduction, to guarantee baseline performance, we can add the acoustic score and language model score from the HMM system. As noted in [17], the Hidden Conditional Random Field (HCRF) model takes on the log-linear form, and in that case our models can coincide. An HMM decoding can produce the n top most likely hypotheses, called n-best list. This forms an ordered list, in which the position of a hypothesis is called its rank. By using Bayes rule, we can also find the posterior probability of each hypothesis, just called posterior, defined as the probability of the hypothesis given the audio, as opposed to the likelihood of the audio given that the hypothesis was meant, which is the raw acoustic score from the HMM. In practice, we use the log-posterior of the HMM instead of the raw scores, or the rank of the hypothesis in the HMM-generated n-best list. These have a smaller dynamic range than acoustic and language model scores, are more comparable to binary values, and are easier to regularize.

B. Nearest Neighbor Features with Dynamic Time Warping

Nearest-neighbor features guarantee optimal performance in the presence of infinite amounts of data. They relate the closest training exemplar (h_t, X_t) to a test utterance X. For sequence input, Dynamic Time Warping (DTW) defines the distance function. DTW-based features tend to be very effective for the most popular queries (so-called "head" data), as they can model idiosyncrasies in pronunciation exactly. However, they do not generalize well.

Dynamic Time Warping is a common technique to calculate the distance between two real-valued sequences of different length, X_{hyp} and X_{tpl}, and the warped distance can be computed efficiently [15], [8]. The complexity of this algorithm is $\mathcal{O}(|X_{tpl}| \cdot |X_{hyp}|)$, *i.e.*, it is basically quadratic in the length of the sequences.

C. Decomposable features

We now turn to features which are able to generalize to unseen words and listings. In a practical system, this is critically important. To get this generalization, one of the techniques we have developed is that of decomposable features. These features are of the form

$\Phi(h)$	$\Psi(X)$	$f(h, X) = \Phi(h)\Psi(X)$
h contains *food*	*food* has been spoken	word spotter

TABLE I
WORD-SPOTTER AS A DECOMPOSABLE FEATURE.

Attributes (word spotters)	
{*chinese, food, mexican, restaurant*}⊂h	
Seen events	Unseen events h covered by word spotters
mexican restaurant	*mexican food*
chinese food	*chinese restaurant*

TABLE II
ILLUSTRATION OF GENERALIZATION ISSUE USING WORD SPOTTERS.

$$f(h, X) := \Phi(h)\Psi(X). \tag{3}$$

The $\Psi(X)$ is called the *acoustic* component, and the $\Phi(h)$ is called the *linguistic* component. A canonical example consists of the product of two binary features: for example, an acoustic detector assessing the presence of a sibilant at the end of the utterance for $\Psi(X)$, and the presence of the letter 's' at the end of the business name for $\Phi(h)$. Another example is illustrated in Table I, for a class of features we refer to as word spotters – features that measure the congruence between a word which we expect to see based on the hypothesis, and an acoustic detection of that word. The acoustic detection may be made via the DTW based approach just described, or with a parametric model.

The way we achieve generalization is by making either $\Psi(X)$ or $\Phi(h)$ coarse. At the fine-grained extreme, one may use a $\Psi(h)$ which triggers for a single business identity. For instance, we may decide to activate features only if *Walmart* is considered as a hypothesis. At the other (coarse) extreme, we may decide to have a feature if h contains two words or less - which fires for about half of utterances. Similarly, for acoustics, a fine-grained feature $\Psi(X)$ might be defined to fire only when X is deemed similar enough to a given example of *Walmart* in the training data, or (coarsely) when the length of X is below 3 seconds. Table II illustrates the generalization that is possible, *e.g.* with word-spotters. Ideally, to classify the 30M businesses in our database, we need 25 features which have perfect mutual information, each splitting the space in half and providing one bit of information. Purely linguistic features, *i.e.*, features for which $\Psi(X)$ is set to a constant, can also be defined. They are language model features, and were used successfully in [20].

To conclude this section, we would like to note that decomposable features are useful not just for their generalization ability, but also because it allows the feature engineer to think of acoustic and linguistic features in isolation. This allows researchers to focus on their expertise, and then to link their work together in a simple and convenient way.

D. Extracting features from a graph of multi-phone units

The example of word-spotters from the previous section utilized word-detectors for $\Psi(X)$. Unfortunately, there are a very large number of words, and it may not be feasible to train a detector for each. Further, if template-based methods are used, we are faced with the generalization problem that test words may be unseen in the training data. Therefore, we are driven to explore the use of subword units. The ideal units will carry a large amount of information about the words, and (necessarily) be reliably detected; Section IV will describe in detail how we select these units. In the remainder of this section, we will assume these detector units as input, and describe the features which are extracted.

As input, we represent the audio with Mel Frequency Cepstral Coefficients (MFCCs) and then turn it into a graph of multi-phone units. Examples of these units are /ae_r_ax_n/ (as in Aarons) and /th_iy/ (as in Abern*athy*). The multi-phone units $\{u\}$ are then associated with words $\{w\}$, for example via a lexicon specifying the linkage, or simply from co-occurrence statistics. In effect, we break up the statistical dependency from audio input X to output h as the Markov chain:

$$X \rightarrow \{u\} \rightarrow \{w\} \rightarrow h. \tag{4}$$

In other words, we assume the information provided by X is entirely contained in $\{u\}$ for the purpose of finding words, and in turn they are sufficient to decide $\{w\}$, then h. At this stage, however, we do not imply that large units are a concatenation of small units or acoustic frames, merely that larger units depend on smaller units, for example through co-occurrence This is in contrast with HMMs, which follow a similar hierarchical construction, in that:

- HMMs typically use triphones as $\{u\}$, which have a short span;
- the pronunciation dictionary performs a near-deterministic mapping from $\{u\}$ to $\{w\}$; and
- the links in the Markov Chain must be Markovian with respect to the underlying units, which is the primary source of computational efficiency.

We distinguish two ways of defining features, both of which are violated the Markovian property of HMMs. Our features are based on Eq (4), 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. Both of these features make use of quantities defined by a simple model for determining the probability of a hypothesis. The model assumes that units are conditionally independent given X, distributed according to counts observed in the lattice. Further, we assume that each word is produced independently, given a unit, again independently of other units in the bag. The model is:

$$p(h|X) = \sum_{u,w} p(u|X)p(w|u)p(h|w), \tag{5}$$

Name	$\Psi(X)$	$\Phi(h)$	N
uw	$\gamma(u\|x)p(w\|u)$	$\delta(w \in h)$	10^6
word	$\sum_u \gamma(u\|x)p(w\|u)$	$\delta(w \in h)$	30k
hyp	$f(h, X) = \log p(h\|X)$		1

TABLE III
DEFINITION OF ASSOCIATIVE FEATURES OF EACH TYPE; N IS THE TYPICAL NUMBER OF FEATURES DEFINED FOR EACH TYPE.

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

TABLE IV
RECOGNITION ERRORS WITH ASSOCIATIVE FEATURES.

with a deterministic mapping of h to words:

$$p(h|w) = \frac{p(h)}{p(w)}\delta(w \in h), \tag{6}$$

and $\delta(\cdot)$ is the indicator function. Note that u and w here are individual units and words - not sequences, and in contrast with a standard HMM, $p(w|u)$ is not determined by a pronunciation lexicon: rather, it is a co-occurrence 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 classes of associative and transductive features are now defined.

1) Associative features: The definition of associative features is presented in Table III. In the third column of that table, as N, we record a typical number of features in our task. The vocabulary size of 30k defines how many "word" features we have. On average, each word is seen in the same utterance with 33 distinct units, so there are $33 \times 30k = 10^6$ uw features. These features simply associate the presence of a multi-phone in the decoded unit graph with a given output or word. The features are "Pavlovian": given a certain multi-phone stimulus, a response in the form of a certain output can be trained. They rely entirely on co-locations of units with expected output words in the training data. They are similar in spirit to Episodic Memory models [21], albeit starting from discrete symbols rather than acoustics. Associative features also bear a relationship to Bellegarda's Latent Semantic Analysis [22], which is based on co-occurrences of words in documents. The generative model of Eq (5) is added as a feature $f(h, X) = \log p(h|X)$.

Associative features are global in nature: it is not necessary that a unit representing a certain sound is contained within the dictionary pronunciation of a word for them to be considered together. They also do not consider ordering constraints.

In early stages of development, we noticed that this property causes some interesting mis-recognitions, as shown in Table IV. 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.

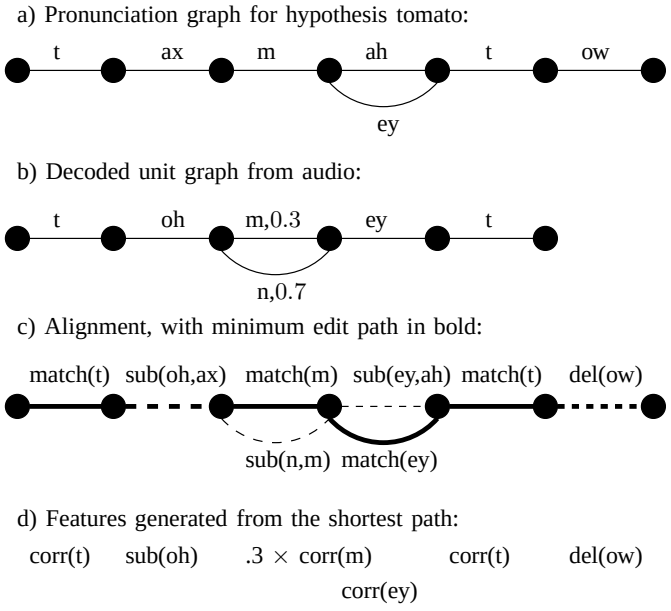

Fig. 2. Transductive features. The posterior for units m and n are 0.3 and 0.7 respectively. Edit operations along the minimum edit path are recorded as transductive features.

Name	$f(h, X)$	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 V
TRANSDUCTIVE (LEVENSHTEIN) FEATURES ARE EXTRACTED FROM LEVENSHTEIN ALIGNMENT. N IS THE TYPICAL FEATURE COUNT OF EACH TYPE.

2) Transductive features: With transductive features, we assume the additional knowledge of a pronunciation dictionary. Thus, we have a prejudice about which units should appear in the audio and in which order, to be consistent with an output hypothesis h. The process for computing transductive features is illustrated in Figure 2. First, we create a pronunciation graph by enumerating the possible hypothesis pronunciations in a lattice structure. We then generate an alignment graph which is the composition of the pronunciation graph with the decoded unit graph, with posterior weights removed. Then, we compute the path with minimum edit distance in this alignment graph. If multiple paths achieve the same minimum edit distance, we pick the one associated with units decoded with maximum posterior. Then, transductive features are extracted by recording edit events following that path. When encountering a substitution event, we ignore the identity of the hypothesis pronunciation unit – in the figure, sub(oh,ax) becomes sub(oh). The feature value is set to the posterior of the arc in the decoded input graph.

Transductive features are enumerated in Table V. Each edit operation (ins, del, sub, corr) is defined once per unit: N, the feature count, is equal the number of units. Transductive features are not decomposable, but they do generalize well since they are defined at the sub-word unit level.

In Table VI, we provide an example utterance and extracted features. The decoded unit graph has three units, "ax p_iy_t_s_ax b_ay". We show two possible hypotheses, Pizza Hut and Best Buy. The associative features associate units with words, as seen in the training set. Note that associative features will only be present for word-unit combinations seen in the training data. In this example, we assume "pizza" and "b_ay" have not been seen together in the training data. Likewise, because "ay" was often wrongly decoded in as "ax", we have an associative feature for (Buy, ax). This mechanism can learn arbitrary pronunciations, if given enough examples. The transductive features, informed by the pronunciation dictionary, can indicate, for instance, that "ax" is an extraneous sound in both cases.

Decoded units	ax p_iy_t_s_ax b_ay
Hypothesis	Pizza Hut
Associative features	(Pizza, p_iy_t_s_ax)
	(Hut, p_iy_t_s_ax)
Transductive features	insert(ax)
	match(p_iy_t_s_ax)
	substitute(b_ay)
Hypothesis	Best Buy
Associative features	(Best, b_ay)
	(Buy, b_ay)
	(Buy, ax)
Transductive features	insert(ax)
	match(b_ay)
	substitute(p_iy_t_s_ax)

TABLE VI
AN EXAMPLE OF ASSOCIATIVE AND TRANSDUCTIVE FEATURES FOR TWO HYPOTHESES (*Pizza Hut* AND *Best Buy*). THE DECODED UNIT GRAPH WAS A SINGLE PATH.

IV. MMI MULTI-PHONE DETECTORS

The features which we have just described have been based on the detection of subword units in the audio. Now, in this section, we present a method of identifying a set of highly informative multi-phone units to use as the basis for those detectors. Our units are similar to syllables [23], but in contrast to previous work [24], [25], they are identified using a discriminative rather than Maximum Likelihood criterion.

We begin by noting that the mutual information between phone sequences and words is a function both of the dictionary pronunciation of words, and of our ability to detect the phone sequences. From the dictionary point of view, a phone sequence occurring in half the words is ideal (providing one bit of information), while one occurring in every word (or no word) is uninformative. From the detection point of view, a sequence which cannot be detected with any reliability is also uninformative, even if it would be otherwise ideal. Key to computing mutual information is therefore the way in which we represent our ability to detect units, and this section is organized around different methods for doing so. The concept of MMI multi-phone units was first introduced by us in [16], and after recapitulating it here, we present new error models in Sections IV-D and IV-E, along with a new method for exactly and efficiently computing the mutual information in Section IV-F. New comparisons of these methods are presented later in this section and in the experimental results.

First, we define the mutual information between phone sequences and words, and analyze it in the errorless case. Then, we consider the use of two different detection error models: one in which the probability of an error is a function of the length of a unit (longer units being more reliably detected), and a much more detailed one in which the probability of an error is empirically derived as a function of the unit identity itself. Both of these error models permit highly efficient routines for computing the mutual information between phone sequences and words. The runtime of the first algorithm is proportional to the size of the dictionary; the runtime of the second is proportional to the size of the dictionary plus the size of a set of phone decodings used to estimate the error model. Both of these result in a mutual information quantity for each phone sequence that occurs in the dictionary. Finally, we show how the mutual information can be computed exactly, using a set of phone decodings to encapsulate the detection uncertainty. The runtime of this algorithm is much longer - proportional to the vocabulary size times the number of distinct phone sequences that occur in the set of phone decodings. However, we present an algorithm that achieves this runtime using only $O(U)$ RAM, where U is the number of extracted units, thus making the exact computation feasible. Experimental results are presented for these various cases in Section VI.

A. Definition

Consider a multi-phone unit u. We use $U = \{0, 1\}$ to denote the presence or absence of the multi-phone unit u. We define a second random variable, W, which can take on the identity of a word. The mutual information between the presence of a unit u and the words is then given by:

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

We now consider several procedures for computing the mutual information between phone sequences and words. As inputs, we will take a dictionary that indicates the phonetic spelling for each word, along with the unigram counts for each word. Two of the methods further require an unconstrained phonetic decoding of some development data, from which to extract statistics. 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.

B. The Errorless Case

We proceed in the errorless case by breaking the set of words up into those in which u is present ($\mathcal{W}^+$) and those in which it is not present ($\mathcal{W}^-$). In the remainder, the word w^+ will be always understood to be summed over $\mathcal{W}^+$, and similarly w^- will always drawn from $\mathcal{W}^-$. We may take advantage of the fact that $P(u=1|w^-) = 0, \forall w^- \in \mathcal{W}^-$ and $P(u=0|w^+) = 0, \forall w^+ \in \mathcal{W}^+$, to simplify:

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

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, 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.

C. 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. 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:

$$\begin{aligned} &MI(U;W) \\ &\quad = \sum_{w} P(w)P(u=1|w) \log \frac{P(u=1|w)}{P(u=1)} \\ &\quad + \sum_{w} P(w)P(u=0|w) \log \frac{P(u=0|w)}{P(u=0)} \\ &\quad = \sum_{w^+ \in \mathcal{W}^+} P(w^+)P(u=1|w^+) \log \frac{P(u=1|w^+)}{P(u=1)} \text{correct accept} \\ &\quad + \sum_{w^+ \in \mathcal{W}^+} P(w^+)P(u=0|w^+) \log \frac{P(u=0|w^+)}{P(u=0)} \text{false reject} \\ &\quad + \sum_{w^- \in \mathcal{W}^-} P(w^-)P(u=1|w^-) \log \frac{P(u=1|w^-)}{P(u=1)} \text{false accept} \\ &\quad + \sum_{w^- \in \mathcal{W}^-} P(w^-)P(u=0|w^-) \log \frac{P(u=0|w^-)}{P(u=0)} \text{correct reject} \end{aligned}$$

When we have an error model $P(u = \{0,1\}|w^{\{+,-\}})$ that is only a function of the unit u, this simplifies further because the sum can be factored out:

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

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 is examined, and the mutual information computed according to the above formula. The necessary quantities are readily available:

- $\sum_{w^+ \in \mathcal{W}^+} P(w^+)$ and $\sum_{w^- \in \mathcal{W}^-} P(w^-)$ are computed in the initial pass over the data for each u as in the errorless case.
- $P(u=1|w^-) = f(u)$, where $f()$ is a function that estimates the false positive probability
- $P(u=0|w^-) = 1 - P(u=1|w^-)$

Unit: Length Model (A)	$MI(U;W)$	Unit: Length Model (B)	$MI(U;W)$	Unit: Empirical Model	$MI(U;W)$	Unit: Exact	$MI(U;W)$
ax_n	0.026 bits	ax	0.29	r	0.41	r	0.44
k_ae_l_ax_f_ao_r_n_y_ax	0.023	r	0.29	n	0.32	ax	0.38
ae_l_ax_f_ao_r_n_y_ax	0.022	n	0.27	ax	0.21	n	0.37
k_ae_l_ax_f_ao_r_n_y	0.022	s	0.25	l	0.28	l	0.32
l_ax_f_ao_r_n_y_ax	0.022	t	0.25	s	0.28	s	0.32

TABLE VII
THE MOST INFORMATIVE MULTI-PHONE UNITS FOR THE DIFFERENT METHODS OF ESTIMATING MUTUAL INFORMATION. AN "_" IS USED BETWEEN THE PHONES BELONGING TO A SINGLE UNIT.

- $P(u=0|w^+) = g(u)$, where $g()$ is a function that estimates the false negative probability
- $P(u=1|w^+) = 1 - P(u=0|w^+)$
- $P(u=1) = P(u=1)\sum_{w^+} P(w^+) + P(u=1)\sum_{w^-} P(w^-)$
- $P(u=0) = P(u=0)\sum_{w^+} P(w^+) + P(u=0)\sum_{w^-} P(w^-)$

Note that in the last two bullets we have taken advantage of the fact that $P(u=k|w^{\{+,-\}}) = P(u=k), \forall k \in \{0,1\}, w^+ \in \mathcal{W}^+, w^- \in \mathcal{W}^-$ is only a function of u to move this factor outside the summations.

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 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. With the exact mutual information computation of Section IV-F, no approximations are made.

D. Length-Based Error Model

The simplest of our approaches uses a length-based error model to implement $f()$ and $g()$. In this model, we capture the intuition that the likelihood of hallucinating a specific unit will fall off rapidly as the length of the unit increases. The number of distinct units with a given length increases exponentially with the length, and so it is reasonable to assume that the probability of falsely accepting a specific one will be exponentially decreasing in the length of the units. The length-based model further captures the observation that the probability of falsely rejecting word units is roughly constant, regardless of length. We thus define:

- $f(u) = P(u=1|w^-) = ae^{-bl}$ where l is the length of the unit in phones, and a and b are constants.
- $g(u) = P(u=0|w^+) = c$, a constant

We have experimented with two methods for setting the constants. In the first, which we will call method A, we simply use $a = 1$,$b = 1$, and $c = 0.5$. This overestimates the effect of errors, as it is unlikely that a single phone unit will be falsely detected with probability $\exp(-1) \approx 0.37$. The second method, which we will call method B, follows a more careful line of reasoning. Let us assume that the majority of errors that we would see in an edit distance computation are substitutions, and that each is manifested by a false rejection of the correct unit, and a false acceptance of the incorrect one. For phone-based recognition, where the unit length is 1, a 30% error rate is reasonable. At the word level, where the unit length is about 6 on average, we expect a 45% error rate. We will further assume that false positives are equally distributed over the space of possible units - 40 units in the case of phones, and around $100,000$ in the case of words. We may then solve the equations:

$$a\exp(-b) = \frac{0.3}{40}$$
$$a\exp(-6b) = \frac{0.45}{100,000}$$

This results in $a = 0.03, b = 1.5$. Finally, we note that the false negative rate will just vary between 0.3 and 0.45, and we may use the average as a reasonable value. In subsequent tables, we will present results for both settings of the model parameters, and compare with more exact computation of the mutual information.

Unit	$MI(U;W)$
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 VIII
THE MOST INFORMATIVE MULTI-PHONE UNITS AFTER UNIT SELECTION USING THE LENGTH (A) BASED ERROR MODEL. FOR OTHER MODELS THE LISTS ARE UNCHANGED.

E. Empirical Error Model

The simple length-based error model above may miss important phenomena, such as the fact that detecting a plosive like /t/ may be more difficult than detecting a vowel like /ah/. To capture more detailed information like this, we have developed an empirically derived error model. This model is based on doing an unconstrained phone decoding, and then using the time marks from a forced alignment of the transcription to extract the phones in the unconstrained decoding which are associated with each word. When the forced alignment is done, a single lexical variant of each word is selected from among the pronunciations possible. Thus, we may associate with each actual phonetic realization of a word from the unconstrained decoding an expected phonetic realization from the forced alignment. For any particular multi-phone unit, we can then determine if it was falsely accepted or falsely rejected. We then use these empirically determined quantities for the $f()$ and $g()$ functions:

- $f(u) = P(u=1|w^-); \forall w^- \in \mathcal{W}^-$ (false accept)
- $g(u) = P(u=0|w^+); \forall w^+ \in \mathcal{W}^+$ (false reject)

This approach has the benefit of allowing for unit specific probabilities, with the drawback that we may obtain unreliable estimates for very infrequent units.

Word	Unit Breakdown: Length (A)	Breakdown: Length (B)	Breakdown: Empirical	Breakdown: Exact
Academia	ae_k_ax d_iy m_iy ax	ae_k_ax d_iy m_iy ax	ae_k_ax d_iy m_iy ax	ae_k_ax d_iy m_iy ax
Academic	ae_k_ax d_eh m_ih_k	ae_k_ax d_eh m_ih_k	ae_k_ax d_eh m_ih_k	ae_k_ax d_eh m_ih_k
Academics	ae_k_ax d_eh m_ih_k s	ae_k_ax d_eh m_ih_k s	ae_k_ax d_eh m_ih_k_s	ae_k_ax d_eh m_ih_k_s
Academies	ax_k_ae_d_ax_m_iy z	ax_k_ae_d_ax_m_iy z	ax_k_ae_d_ax_m_iy z	ax_k_ae_d_ax_m iy_z
Academy	ax_k_ae_d_ax_m_iy	ax_k_ae_d_ax_m_iy	ax_k_ae_d_ax_m_iy	ax_k_ae_d_ax_m iy

TABLE IX
SEGMENTATION OF SEVERAL WORDS INTO MULTI-PHONE UNITS.

F. Exact Mutual Information Computation

To do the exact mutual information computation, we directly implement the definition:

$$MI(U;W) = \sum_w P(w)p(u=1|w)\log\frac{P(u=1|w)}{P(u=1)} + \sum_w P(w)P(u=0|w)\log\frac{P(u=0|w)}{P(u=0)}, \quad (7)$$

and exhaustively enumerate $P(u=\{0,1\}|w)$ for all words and units.

In common with the previous method, we use a set of unconstrained phone decodings to represent our ability to accurately decode phone sequences. However, the computations must be carefully organized. If we denote the vocabulary size by V and the number of units by U, in the exact computation, a runtime of $O(VU)$ is inescapable. However, a much smaller memory requirement of $O(U)$ is possible (we refer to RAM; file storage to hold the data is taken as given). In fact, were this not possible, the computation would be prohibitive due to memory use.

First, we examine the set of decodings and collect a set of candidate multi-phone units, those that are shorter than a given length (≤ 12 in our experiments), and those which occur at least a certain number of times (10). The number of times each unit is seen is stored. We also tabulate the number of times each word is seen. The mutual information is then computed as follows:

1) Extract the phonetic realization of every word in the training data
2) Store this in a file with two columns: the word in one and the realization in the other. There is a line for each word occurrence.
3) Sort the file on words.
4) Scan through the file sequentially, and for each word's realizations:
   a) Initialize $P(u=1|w)=0\ \forall u$ (Note that since we examine the words sequentially, we know w and need only use a container the size of $|U|$.)
   b) Examine each realization of w and for each candidate sequence present in the realization, update $P(u=1|w)$.
   c) After all realizations of the current word have been scanned, for each unit:
      - Compute $P(u=0|w)$ as $1-P(u=1|w)$.
      - Update $MI(U;W)$ according to Eq (7).

G. Unit Selection

Table VII shows the five most informative multi-phone units for our four methods of computing mutual information.

Method	Number of Units
Length (A)	4358
Length (B)	4578
Empirical Error Model	4679
Exact MI	4601

TABLE X
NUMBER OF UNITS AFTER UNIT SELECTION.

Note that the mutual information quantities listed depend on the error model used (an error model specifying completely random detection would result in 0 mutual information), and therefore the columns are not expected to agree. In length model A, the coefficients $a=1, b=1$ have overestimated the probability of obtaining false-positives with short phone sequences, with the result that these do not appear amongst the top-5 most informative units. Instead, much longer units are present for this length model. As can be seen, many of these 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 of one unit given another is low.) While it is not apparent from the list of the top-5 units, this same issue arises with the other methods of estimating mutual information as well - the units involved just appear lower down on the list. Therefore, we are motivated to develop a procedure for removing redundancy from the list.

To do this, we proceed by defining a set of candidate units - the top $N=10,000$ most informative - and then partitioning each dictionary word into the minimum number of candidate units. Any unit that is selected fewer than 50 times is then thrown away. The most informative of the selected units are shown in Table VIII. To get a sense of how the units are used, Table IX shows the segmentation of several words. This illustrates how words of modest frequency are typically decomposed into syllable-like and single-phone units. Table X shows the number of units present after unit selection for the different methods. Table XI shows the segmentation of the most common business requests, and illustrates the fact that very common words typically result in whole-word units. For the most common words, all methods produce the same segmentation, with the exception of an unusual pronunciation of "McDonald's;" the variability here is shown in Table XII. To get a sense of the variability present in the segmentations, we have tabulated the fraction of identical units for the different methods. This is illustrated in Table XIII. We see that there is a significant amount of similarity between the results.

Word	Unit Breakdown
Pizza	p_iy_t_s_ax
Wal-Mart	w_ao_l_m_aa_r_t
McDonald's	m_ih_k_d_aa_n_ax_l_d_z m_ax_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 XI
SEGMENTATION OF THE MOST COMMON REQUESTS INTO MULTI-PHONE UNITS WITH THE EXACT MUTUAL INFORMATION COMPUTATION.

Method	McDonald's 1	McDonald's 2
Length (A)	m_ih_k_d_aa_n_ax_l_d_z	m_ax_k_d_aa_n_ax_l_d_z
Length (B)	m_ih_k_d_aa_n_ax_l_d_z	m_ax_k_d_aa_n_ax_l_d_z
Empirical Error	m_ih_k_d_aa_n_ax_l_d_z	m_ax_k_d_aa_n_ax_l d_z
Exact MI	m_ih_k_d_aa_n_ax_l_d_z	m_ax_k d_aa_n_ax_l_d_z

TABLE XII
SEGMENTATION OF "MCDONALD'S" INTO MULTI-PHONE UNITS. ALL METHODS GIVE THE SAME ANSWER FOR THE COMMON PRONUNCIATION, THOUGH THERE IS VARIATION IN THE MORE UNUSUAL PRONUNCIATION.

H. Qualitative Analysis of Units

An ideal unit from the mutual information standpoint would occur in exactly half the words. In fact, no unit occurs nearly this often, and the units that come closest to this ideal are single-phone units. In the absence of errors, single-phone units would be best, followed by two-phone units and so forth, in order of decreasing frequency. However, imperfect detection counteracts this: single phone units are more likely to incur false-accepts and therefore they are penalized by the mutual information criterion. In the final set of units, we see both some very short units such as "ax n," which are present due to their frequency, and also some very long ones such as "k ae l ax f ao r n y ax," which are present due to their assumed detection reliability (and relative frequency). The net result of this process is a set of multi-scale units with which words can be represented. The scale varies from single phone units to syllable-like and whole-word units, and the determination of the set of units to work with is made with a sliding scale that uses mutual information to trade off frequency with detection reliability.

V. LIMITATIONS

In this section, we contrast differences with the HMM approach and limitations in the system proposed in this paper. First, we look at the model itself. Then, we examine potential limitations with the features we have chosen. Finally, we delve into more issues which arise in practice.

A. Modeling

FDMs do not impose the frame-level Markov assumption. Their log-linear form implies that there exists a linear decision boundary. More complex decision boundaries must be implemented by adding more complex features. In practice, we have a large number of features, and we do not envision this to be the main problem. Regularization with a single prior weight further implies that feature scaling is important.

In effect, the simple log-linear form of FDMs delegate modeling assumptions to feature design.

	Length (A)	Length (B)	Empirical	Exact
Length (A)	-	91%	77	68
Length (B)		-	78	72
Empirical			-	77

TABLE XIII
FRACTION OF DERIVED UNITS IN COMMON BETWEEN METHODS.

B. Features

In practice, the main limitations in HMMs which FDMs can target are: frame-level Markov decisions, a nearly deterministic pronunciation dictionary, and state-tying. Each feature type must be examined separately.

1) Dynamic Time Warping: These features learn to map example sequences of audio to a hypothesis. Like HMMs, they operate on a frame-by-frame basis and the final score is a sum/product of frame similarity scores. There is no notion of state-tying or pronunciation dictionary. The main drawbacks are that they require large amounts of data to train, and are computationally expensive.

2) Associative features: While they rely on HMM-generated input, associative features make the fewest assumptions. From Eq (4), it is assumed that a hypothesis is composed of a bag of words, and that words are constructed from a bag of units from the entire audio sequence. These features do not use a pronunciation dictionary. They cannot distinguish word order or word boundaries between units.

Associative features work globally at the utterance level. As seen on Table IV, it is possible to suggest words which have not been realized acoustically. In the case of business search, it performs the function of correction, for instance, possibly suggesting "Fred Meyer" in lieu of an incorrect "Fred Meyers".

3) Transductive features: Using a pronunciation dictionary, transductive features exploit word and unit order information to perform matching. They are the closer to HMMs. They consider multi-phone errors in isolation. In effect, they allow training of a context-free multi-phone edit model. In an isolated word recognition context, a similar effect can be obtained with HMMs by introducing many pronunciations variants, each penalized by edit operations which deviate from the closest canonical pronunciation. Therefore, in an HMM implementation, they may be thought of as a tied model for discriminative lexicon training.

The feature set should be considered in its entirety: the role of each feature is to add complementary information to the other features. Therefore, a specific limitation of a particular feature will decrease its own effectiveness, but it could be captured by another feature. In general, if phenomena may be quantified independently, FDMs will suffer no loss.

C. Practical considerations

Three fundamental limitations arise from the configuration in which the FDM was implemented.

1) Task dependence: HMMs are based on a decomposition of acoustics, language model (the channel model), and lexicon. Very early on and aggressively, FDMs assume a single task. It

	Length(A)	Length(B)	Empirical	Exact
Data required	-	Estimates of phone and word error rates; Word length	Phone decode	Phone decode
Intermediate model	parametric $f(), g()$	parametric $f(), g()$	Edit distance tables	-
Runtime	$O(U+V)$	$O(U+V)$	$O(U+V)$	$O(UV)$

TABLE XIV
COMPARISON OF ERROR MODELS IN TERMS OF IMPLEMENTATION.

is not possible to deploy FDMs trained in a given task, with a given vocabulary, to a new task. In an HMM system, one may replace the language model and lexicon and keep the acoustic models.

2) Speaker adaptation and noise robustness: Beyond using modified input HMM system, it is not immediately clear how to perform speaker adaptation, or apply signal-level noise robustness techniques in FDMs. In our work, we assume features derived from multi-phone units and words. Thus, speaker adaptation, for instance, would be akin to lexicon and language model adaptation, not acoustic adaptation.

3) Features are unrelated: HMMs follow a single consistent generative framework. By contrast, our features are conceptually unrelated to each other, even when they capture similar phenomena. Therefore, advances in the generation of one feature may not automatically benefit other features.

VI. EXPERIMENTS

In our experiments, we try to elucidate several questions regarding the effectiveness of the approach, namely:

- Q1. Can we improve on an HMM baseline with nearest-neighbor units, which only improve a few cases, but do not generalize?
- Q2. Does the FDM, on its own and without the HMM baseline, decode with acceptable accuracy?
- Q3. What is the effect of the multi-phone error model on recognition results in the FDM?
- Q4. Does knowledge from the pronunciation dictionary help through the use of transductive features?
- Q5. In practice, can unit-based features complement an HMM system and improve accuracy?

First, we describe the general experimental framework. Then, we proceed to describe Nearest-Neighbor and multi-phone experiments.

A. Bing Mobile Voice Search Data

For our experiments, we consider the Voice Search task. We have deployed an application [9] called Bing Mobile, which allows users to look for local businesses from their mobile phone. Speech comes in various challenging conditions, including outside noise, music, side-speech, sloppy pronunciation, and different acquisition channels. Key to our acquisition of training data, once a query is spoken, a list of alternatives is presented for user validation. This has resulted in a training set of approximately 3M utterances. By comparing user selections with transcriptions for a small set of transcribed data, we estimate that the user provided data is 85-90% accurate. Our test data consists solely of the more accurately transcribed data. Table XV summarizes the training set and test set configurations used for experiments. The nearest neighbor experiments were performed early on, with the training and test data listed. For our later multi-phone experiments, we used more training data and two test sets. The first (full test) consists of 8777 business utterances collected in the week of May 16, 2008. The second is a subset of 3623 of these, which correspond to requests of one of the 1000 most frequently requested businesses. All HMM baselines are trained using the maximum-likelihood criterion (ML), rather than discriminatively, like the FDM. We believe that we would still see improvements over a discriminatively-trained HMM baseline.

Nearest Neighbor experiments	
Training set	550k utterances
Test set (Table XVII)	21k utterances
HMM baseline (SER)	39.6%
HMM 100-best oracle (SER)	21.6%
Multi-Phone experiments	
Training set	3M utterances
Test set (Table XIX)	8777 utterances
HMM baseline (SER)	34.7%
HMM 100-best oracle (SER)	16.8%
Test set (Table XVIII)	3623 utterances (top 1000 only)
HMM baseline (SER)	13.4%
HMM 100-best oracle (SER)	1%

TABLE XV
CONFIGURATIONS USED IN THE EXPERIMENTS.

In general, during decoding, FDM runs at a fraction of HMM decoding. Training is also relatively expedient. For example, ML training of HMM models on 2000 hours with 1024 CPUs takes about 7 hours, while FDM training always finishes within 3 hours for the most expensive case, and otherwise typically within 1 hour. The dominating cost for FDM is the generation of multi-phone decodings or DTW matching, *i.e.* for feature extraction.

B. Nearest-Neighbor Experiments

Our first set of experiments, reported in [15], addresses the question of whether nearest-neighbor features, which target the most frequent listings (so-called "head" events), can improve upon an HMM baseline system (Q1). It is important to demonstrate that FDM can make use of features or models that, by design, only target specific phenomena or listings, but may not generalize. In practice, it allows researchers to find new alternative models without having to design them to work on all utterances, and to focus on specific areas of strength. Here, we examine what gains can be had from using Nearest-Neighbor features from Section III-B. The HMM system

Name	$\Psi(X)$	$\Phi(h)$	N
letter 6-gram	1	$\delta(6gm \in h)$	50k
DTW	distance to closest example of c	$c \in h$	3-15k
posterior	$f(h, X) = \log p(h\|X)$		1
rank	$f(h, X) =$ n-best rank of h		1

TABLE XVI
FEATURES USED IN TEMPLATE EXPERIMENTS. N IS THE FEATURE COUNT.

Setup	SER [%]	
	Actual	Total
HMM baseline	17.4	39.6
With NN features	13.0	36.3

TABLE XVII
RESULTS IN SENTENCE ERROR RATE (SER) WITH NN FEATURES.

operated on MFCC/HLDA 36-dimensional features, had 3,000 tied-state mixtures and 16 Gaussians per mixture and yielded 39.6% sentence error rate (SER). The system generated the candidate list for the FDM. All features were trained on a subset of 550k utterances, as described in [15].

The features used in these experiments are summarized in Table XVI. The number of NN features depends on whether utterance or word-level spotters are used. The letter 6-gram is a language model feature. The number of features N was chosen to keep experiments manageable. Table XVII shows results obtained with the NN approach. The second column, "Actual", reports the error rate on the utterances for which the reference transcript was found in the n-best list. In the third column, "Total", the full test set was considered including all utterances. The first line corresponds to the baseline error rate, which can be obtained by just incorporating feature in line 4 of Table XVI. To this, we added the NN features and other features from Table XVI. We observe a reduction of roughly 4% in error rate from the HMM baseline.

C. Multi-phone unit experiments

To explore the use of word and sub-word units, we now turn to the use of multi-phone units. We would like to know whether units extracted in Section IV are a good discrete summarization of the audio for the purpose of recognition (Q2).

1) Recognition using units: First, we would like to know whether features based on the hierachical unit-word-hypothesis decomposition in Eq (4) and Eq (5) is comparable with the state-of-the-art HMM. If units contain sufficient information and if our model is reasonable, one would expect results comparable with HMM.

To determine this, we use a setup similar to [16]. There are 3M utterances of training data, divided into two sets: one for feature training (hyper-parameters), and one for model training (lambda parameters). Note that, given Figure 1, we would need a 100M hypothesis space to achieve an oracle rate of 80%. In our implementation, that exceeded memory capacity. Instead, we performed a set of experiments with the 1000 most common businesses only. It would be inappropriate to generate n-bests from the HMM system, since they introduce a bias. The search over hypotheses was done by enumerating those hypotheses with appreciable probability according to Eq (5).

	Length (A)	Length (B)	Exact	Empirical
1. Associative features	12.72	12.74	12.32	12.24
2. + Terminal S	11.74	11.80	12.13	12.02
3. + Transductive	9.52	9.86	9.95	9.98

TABLE XVIII
ERROR RATES WITH DIFFERENT UNIT SELECTION TYPES AS FEATURES CLASSES ARE ADDED TO THE SYSTEM. TOP-1000 TEST SET. A STATE-OF-THE-ART HMM SYSTEM ACHIEVES 13.4%.

For comparison, a state-of-the-art HMM system on this task would yield 13.4% sentence error rate (SER).

Results are shown on Table XVIII. We will detail each line later, but we can already gather that the FDM, starting from unit decoding graphs, can produce results that outperform the HMM baseline.

2) Effect of the error model (Q3): In the same table, the first line shows results obtained from a system trained on features described in Table III. We can see that the exact MMI criterion of Section IV is a good proxy for recognition by association – the presence of MMI units is a good indicator for the presence of words. In line 2, we add a special kind of feature: it triggers when a given unit is seen as the last unit chronologically (for $\Psi(X)$), and if hypothesized text ends in an 's' (for $\Phi(h)$). Note that, after combination with this feature, Length(A) features now outperform more exact MMI computation features. The MMI criterion optimizes for associative features only, and is now sub-optimal for this configuration. There is no obvious way to characterize why the Length-based multi-phone units are more suited for this configuration. We defer to future work the integration of the terminal S feature into the MMI computation.

3) Adding pronunciation knowledge (Q4): In line 3, we add the transductive features of Table V. In effect, we are making use of causal information (in which order units appear in the audio) and prior lexical knowledge in an explicit way. Associative features do not have a concept of "missing" or "extra" units, nor one of sequence. Note that the pronunciation dictionary was, however, used during the design of the multi-phone units. Our transductive features provide a very large improvement, and overall, we achieve an error 28% relative lower than the HMM system.

4) Improving the state-of-the-art HMM (Q5): The multi-phone results so far have not used the output of the HMM baseline system. Now, we test whether multi-phone units provide extra information which can be used to complement an HMM system. To that end, we run rescoring experiments in which N-best lists generated by an HMM for the full test set are re-ranked according to this model. This setup is analogous to that used in the testing of template features, though with an improved baseline. We have added the log-posterior from the HMM baseline. These results are summarized in Table XIX. We see over 2% absolute improvement on the full test set, indicating that our associative and transductive features convey information not present in the HMM system.

VII. CONCLUSION

In this paper, we have described Flat Direct Models (FDMs). These models are inherently discriminative because they are

Baseline	Length (A)	Length (B)	Exact	Empirical
34.73	32.65	32.63	32.80	32.85

TABLE XIX
ERROR RATES FOR RESCORING THE FULL TEST SET. ALL FEATURES FROM THE PREVIOUS TABLE ARE USED. THE HMM SYSTEM ACHIEVES 34.73%.

conditioned on the input. More crucially, the models do away with the strict requirements of Markovian causality present in HMMs. They are best suited for speech recognition tasks with short sentences and a finite, if moderately large, inventory. In an appropriate embodiment, FDMs are bounded between HMM accuracy and nearest-neighbor accuracy, and depending on amount of data available, the system designer may choose which is the best middle ground.

FDMs are implemented generically as log-linear models, which allow arbitrary features to be incorporated. A good choice of features, therefore, is of supreme importance. We have defined several classes of features. First, the HMM scores can be inserted so as to guarantee HMM performance. Second, with nearest-neighbor features based on DTW alignments, we train optimal non-parametric models on a restricted number of utterances, while retaining baseline accuracy on other utterances. Third, features can be defined as a separable combination of purely acoustic features (based on just the audio) and purely linguistic features (based on just the hypothesized words), for both ease of development and generalization. Fourth, features may be associative when they rely on co-occurrence of acoustics and linguistics, or transductive when they are imparted by a transduction model from audio to words. Most of the features used do not have an equivalent HMM implementation.

An important contribution is the definition of intermediate symbols for representing the acoustics which are most indicative of transcribed word sequence. Multi-phone units are designed to maximize the mutual information between a decoded unit sequence and a target word. Hence, they rely on an error model which mimics errors during decoding. We have experimented with two classes of error models: a length-based error model, which stipulates that longer units are easier to recognize accurately, and a more sophisticated model where errors depend on the unit itself.

In our experiments, we have observed consistent error rate reductions in the range of 2-4% in absolute terms, for both the nearest-neighbor and the multi-phone features, on the voice search for Bing Mobile task.

REFERENCES

[1] S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in Speech Transcription at IBM under the DARPA EARS Program," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, 2006.

[2] M. Gales, D. Kim, P. Woodland, H. Chan, D. Mrva, R. Sinha, and S. Tranter, "Progress in the CU-HTK Broadcast News Transcription System," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, 2006.

[3] S. Matsoukas, J.-L. Gauvain, G. Adda, T. Colhurst, C.-L. Kao, O. Kimball, L. Lamel, F. Lefevre, J. Ma, J. Makhoul, L. Nguyen, R. Prasad, R. Schwartz, H. Schwenk, and B. Xiang, "Advances in Transcription of Broadcast News and Conversational Telephone Speech Within the Combined EARS BBN/LIMSI System," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, 2006.

[4] M. Ostendorf, "Moving Beyond the 'Beads-On-A-String' Model of Speech," in *Proc. IEEE ASRU Workshop*, 1999, pp. 79–84.

[5] B. Y. Chen, Q. Zhu, and N. Morgan, "Learning Long-Term Temporal Features in LVCSR Using Neural Networks," in *Proc. ICSLP*, 2004.

[6] H. Hermansky and S. Sharma, "Temporal Patterns (TRAPS) in ASR of Noisy Speech," in *Proc. ICASSP*, 1999.

[7] W. D. Wachter, K. Demuynck, D. V. Compernolle, and P. Wambacq, "Data Driven Example Based Continuous Speech Recognition," in *Eurospeech*, 2003, pp. 1133–1136.

[8] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template-Based Continuous Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1377–1390, May 2007.

[9] A. Acero, N. Bernstein, R. Chambers, Y. 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. ICASSP*, 2007.

[10] "http://www.tellme.com/you."

[11] "http://vlingo.com."

[12] "http://www.google.com/mobile/apple/app.html."

[13] "http://mobile.yahoo.com/onesearch."

[14] H.-K. J. Kuo and Y. Gao, "Maximum Entropy Direct Models for Speech Recognition," in *Proc. ASRU*, 2003.

[15] G. Heigold, G. Zweig, X. Li, and P. Nguyen, "A Flat Direct Model for Speech Recognition," in *Proc. ICASSP*, 2009.

[16] G. Zweig and P. Nguyen, "Maximum Mutual Information Multiphone Units in Direct Modeling," in *Proc. Interspeech*, 2009.

[17] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden Conditional Random Fields for Phone Classification," in *Interspeech*, 2005.

[18] S. Chen and R. Rosenfeld, "A Survey of Smoothing Techniques for ME Models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 37–50, Jan 2000.

[19] M. Reidmiller, "Rprop - Description and Implementation Details," University of Karlsruhe, Tech. Rep., January 1994.

[20] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative Language Modeling with Conditional Random Fields and the Perceptron Algorithm," in *Proc. ACL*, 2004.

[21] V. Maier and R. Moore, "An Investigation into a Simulation of Episodic Memory for Automatic Speech Recognition," in *Proc. Interspeech*, Sep. 2005.

[22] J. R. Bellegarda, "A Multispan Language Modeling Framework for Large Vocabulary Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, 1998.

[23] A. Ganapathiraju, J. Hamaker, J. Picone, M. Ordowski, and G. Doddington, "Syllable-Based Large Vocabulary Continuous Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, pp. 358–366, May 2001.

[24] M. Bacchiani, M. Ostendorf, Y. Sagisaka, and K. Paliwal, "Design of a Speech Recognition System Based on Acoustically Derived Segmental Units," in *ICASSP*, 1996.

[25] R. Singh, B. Raj, and R. Stern, "Automatic Generation of Subword Units for Speech Recognition Systems," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, 2002.