FROM FLAT DIRECT MODELS TO SEGMENTAL CRF MODELS

Geoffrey Zweig and Patrick Nguyen

Microsoft Corporation
One Microsoft Way, Redmond, WA 98052

ABSTRACT

This paper summarizes recent work at Microsoft on the development of novel direct models. The key characteristic of our approaches is the use of long-span segment level features that relate acoustic properties directly to words. In this approach, the frame-level Markov assumption is replaced by the segment level Markov property, allowing us to extract long-span features. A key issue we address is the definition of generalizable features which allow us to model unseen words. We review two recently developed models that have this property: Flat Direct Models (FDMs), and Segmental CRFs (SCRFs). The first operates in a log-linear framework, and uses utterance level features. The second is also a log-linear model, but defines features at the word-segment level. We present new experimental results comparing the two approaches. We find that both show consistent improvements over a baseline system, and that the extra context available to the FDM enables slightly better performance in a rescoring context. This gain comes at the expense of applicability to first pass decoding, for which the SCRF is better suited.

Index Terms— Flat Direct Model, Segmental CRF, Voice Search, Speech Recognition

1. INTRODUCTION

In recent years, direct modeling techniques have enjoyed increasing popularity in both natural language processing [1, 2] and speech recognition [3, 4, 5, 6, 7]. These models have the property that the probability of a state or state sequence $\mathbf{s}$ given some observations $\mathbf{o}$ is modeled directly as $P(\mathbf{s}|\mathbf{o})$ rather than through the application of Bayes rule and specification of a generative model $P(\mathbf{o}|\mathbf{s})$. At their simplest, they are classification models such as maximum entropy models that specify a distribution over class labels given the features. When applied to sequence modeling, more sophisticated methods are necessary, due to the potentially unlimited number of sequences that can be output. To handle this, methods such as Maximum Entropy Markov Models (MEMMs) [1, 8] and Conditional Random Fields (CRFs) [9] have been developed in the NLP area, and applied to speech recognition [3, 4, 5].

This previous work has advanced the field beyond generative HMM modeling [10] by allowing potentially richer kinds of features; however, this has not been fully exploited because past work has retained the frame level Markov property, and used frame-level features and conventional context dependent phonetic states. Recently, we have begun to attack the frame-level Markov assumption as well, first with Flat Direct Models (FDMs) [6, 7] and more recently with Segmental CRFs [11]. We apply these models to the Bing Mobile Voice Search task [12] in which users of a multimodal cellphone application can speak business names, and receive information such as phone numbers and directions. These utterances are typically just a few words long, making it feasible to analyze them either at the utterance level, or at a finer-grained word level.

With FDMs, we take the first approach - the state in this model corresponds to business identity, and the features are extracted from the entire sequence of audio frames. Each feature is of the form $\kappa_i(x, h)$ where x is the audio, h is a hypothesized business name, and i is the feature index. Since we do not in general see all business names in the training data, the set of features must be carefully designed so that model parameters can be learned with one collection of training data, and then generalize to unseen test data; this is fully discussed in Section 4. While careful feature definition can solve the problem of generalization in FDMs, there is a remaining problem of searching over the space of possible hypotheses. For the most frequent business listings, an enumerative approach can be taken; however, for the tail, and for continuous speech recognition in general, it remains to develop an appropriate search strategy.

In the segmental CRF approach, we address the issue of continuous speech recognition with a sequential approach in which dynamic programming can be used in the search over the hypothesis space. Here, the segmental features of the FDM are retained, but applied at the *word* rather than *utterance* level. As we illustrate in Section 3, the resulting model is a CRF in which each state variable is related to a block of observations rather than a single frame.

The remainder of this paper is organized as follows. First, we fully specify the model structures: for FDMs in Section 2, and for SCRFs in Section 3. Both these models use newly developed classes of generalizable features, which are described in Section 4. Section 5 presents new comparative experimental results, and Section 6 provides concluding remarks.

2. FLAT DIRECT MODEL

With Flat Direct Models [6], a set of "consistency features" is defined between a linguistic hypothesis and the underlying acoustics. In contrast to sequential approaches, the linguistic hypothesis is not required to have any explicit structure (e.g. to be a sequence of words). The posterior probability of the linguistic hypothesis is 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

$$P_\Lambda(h|X) = \frac{\exp(\sum_i \lambda_i \kappa_i(x,h))}{\sum_{h' \in N} \exp(\sum_i \lambda_i \kappa_i(x,h'))}. \quad (1)$$

As will be seen in Section 4, we may have millions of features with widely varying numbers of examples of each. Therefore the objective function is regularized with L1 and L2-norm regularization. Denoting the labels of the training data by w_n and corresponding observation sequences by x_n,

$$\hat{\Lambda} = \arg\max_\Lambda \left\{ \sum_n \log P_\Lambda(w_n|x_n) - \nu \sum_i \lambda_i^2 - \tau \sum_i |\lambda_i| \right\}.$$

3. SEGMENTAL CRF

The motivation behind Segmental CRFs is to retain the segment level features which are used with Flat Direct Models, along with the log-linear form of the model, and then to extend the formalism to handle continuous speech recognition. One natural way of doing this is to view the segment-level features as the observation feature functions used in a CRF [9]. To do this, we apply the Markov assumption at the segment rather than frame level and sum over all possible segmentations of the observation stream which are consistent with a word hypothesis. This means that the original "matrix-multiply" inference method of [9] cannot be used. This problem has been addressed for text processing in the work of [13] which uses the term "Semi-CRF" to refer to the fact that the Markov property is applied now at the segment rather than individual observation level. Whereas [13] views this as a problem of determining a constrained labeling of a standard CRF structure, we prefer to view it as one of determining the CRF structure itself. This is illustrated in Figure 1. The top part of this figure shows seven observations broken into three segments, while the bottom part shows the same observations partitioned into two segments. For a given segmentation, feature functions are defined as with standard CRFs.

As noticed in [14], it is possible to represent a CRF using just one type of feature function that involves two adjacent states, and the observations. In the context of a SCRF, a pictorial representation of this is illustrated in Figure 2. This is the form we adopt. We note that while in principle with a conditional model such as a CRF one may use all the observations at any time, in practice we are only interested in features involving a finite and specific span of observations, and it is this specificity which the diagrams are intended to represent.

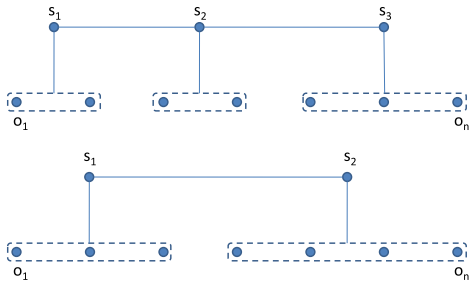

Fig. 1. A Segmental CRF and two different segmentations.

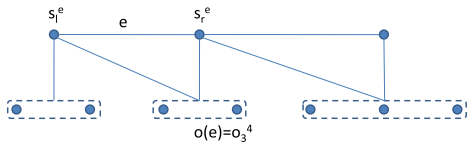

Fig. 2. Incorporating last-state information in a SCRF.

3.1. Model Definition

In the semi-CRF work of [13], the segmentation of the training data is known. However, in speech recognition applications, this is not the case. Therefore, in computing sequence likelihood, we must consider all segmentations consistent with the state (word) sequence **s**, i.e. for which the number of segments equals the length of the state sequence.

Denote by **q** a segmentation of the observation sequences, for example that of Fig. 2 where $|\mathbf{q}| = 3$. The segmentation induces a set of (horizontal) edges between the states, referred to below as $e \in \mathbf{q}$. One such edge is labeled e in Fig. 2 and connects the state to its left, s_l^e, to the state on its right, s_r^e. Further, for any given edge e, let $o(e)$ be the segment associated with the right-hand state s_r^e, as illustrated in Fig. 2. The segment $o(e)$ will span a block of observations from some start time to some endtime, o_{st}^{et}; in Fig, 2, $o(e)$ is the block o_3^4. With this notation, we represent all functions as $f_k(s_l^e, s_r^e, o(e))$ where $o(e)$ are the observations associated with the segment of the right-hand state of the edge. (The first block of observations is treated with an extra notional edge leading into the leftmost state.) The conditional probability of a state sequence **s** given an observation sequence **o** for a SCRF is then given by

$$P(\mathbf{s}|\mathbf{o}) = \frac{\sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}|} \exp(\sum_{e \in \mathbf{q},k} \lambda_k f_k(s_l^e, s_r^e, o(e)))}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}'|} \exp(\sum_{e \in \mathbf{q},k} \lambda_k f_k(s_l'^e, s_r'^e, o(e)))}.$$

Training is done by gradient descent using Rprop [15] and regularization as with FDMs. Taking the derivative of $\mathcal{L} = \log P(\mathbf{s}|\mathbf{o})$ with respect to λ_k we obtain the necessary gradient:

$$\frac{\partial \mathcal{L}}{\partial \lambda_k} = \frac{\sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}|} T_k(\mathbf{q}) \exp(\sum_{e \in \mathbf{q},k} \lambda_k f_k(s_l^e, s_r^e, o(e)))}{\sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}|} \exp(\sum_{e \in \mathbf{q},k} \lambda_k f_k(s_l^e, s_r^e, o(e)))} - \frac{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}'|} T_k'(\mathbf{q}) \exp(\sum_{e \in \mathbf{q},k} \lambda_k f_k(s_l'^e, s_r'^e, o(e)))}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}'|} \exp(\sum_{e \in \mathbf{q},k} \lambda_k f_k(s_l'^e, s_r'^e, o(e)))},$$

with

$$T_k(\mathbf{q}) = \sum_{e \in \mathbf{q}} f_k(s_l^e, s_r^e, o(e))$$

$$T'_k(\mathbf{q}) = \sum_{e \in \mathbf{q}} f_k(s_l'^e, s_r'^e, o(e)).$$

This derivative can be computed efficiently with dynamic programming, using the recursions described in [11].

3.2. Continuous Speech Recognition

In order to model continuous speech, the model structure of Figure 2 is given a specific meaning. While the features we use relate a word to an observation span, the state does not directly encode a word identity. Instead, the values of the state variable in this model correspond to states in a finite state representation of a n-gram language model. This is illustrated in Figure 3. In this figure, a fragment of a finite state language model representation is shown on the left. The states are numbered, and the words next to the states specify the linguistic state. At the right of this figure is a fragment of a CRF illustrating the word sequence "the dog nipped." The states are labeled with the index of the underlying language model state. In our search strategy [11], we extend existing hypotheses with specific words, so the word identity is always available for feature computation.

We use the language model in two ways. First, conventional smoothed ngram probabilities can be returned as transition features. A single λ is trained to weight these features, resulting in a single discriminatively trained language model weight. Secondly, indicator features can be introduced, one for each arc in the language model, which indicate when an arc is traversed in the transition from one state to another. A state transition in the CRF then results in a non-zero feature value (i.e. 1) for each arc traversed in the underlying language model structure. For example, in Figure 3, the arcs $(1, 2)$ and $(2, 6)$ are traversed in moving from state 1 to state 6. Each of these arcs has its own binary feature. Learning the weights on these results in a discriminatively trained language model, trained jointly with the acoustic model.

4. FEATURES

As mentioned in Section 1, we use features that can be trained with one set of words, and then used in cases where other, unseen, words may be present. Our features are based on the detection of phone and multi-phone units [7]. For a given utterance, we form two separate detection streams: one consisting of phones and their detection times, and the other consisting of multiphone units and their detection times. A detection time is a single time associated with a unit, e.g. its midpoint. From each detection stream, several features may be extracted, and these are now discussed in turn. Each is defined with respect to a temporal *span* of detection events and a specific word hypothesis for that span.

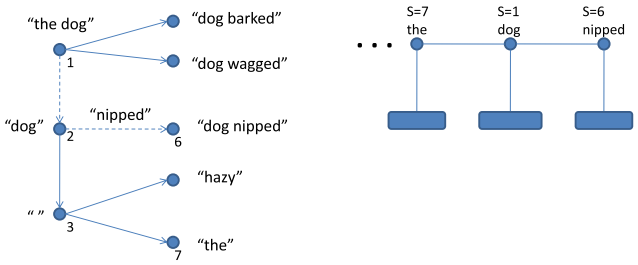

Fig. 3. Correspondence between language model state and SCRF state. The dotted lines indicate the path taken in hypothesizing "nipped" after "the dog." A line from state 7 to state 1 has been omitted for clarity.

4.1. Expectation Features

Expectation features are defined with reference to a dictionary that specifies the spelling of each word in terms of the units. The expectation features are:

- correct-accept of unit u: u is expected on the basis of the dictionary, and it exists in the span
- false-reject of u: u is expected but not observed
- false-accept of u: u is not expected and it is observed

4.2. Levenshtein Features

Levenshtein features are computed by aligning the observed unit sequence in a hypothesized span with that expected based on the dictionary entry for the word. Based on this alignment, the following features are extracted:

- the number of times unit u is correctly matched
- the number of times u in the pronunciation is substituted
- the number of times u is deleted from the pronunciation
- the number of times u is inserted

4.3. Existence Features

Whereas Expectation and Levenshtein features require a dictionary, Existence features indicate the simple association between a unit in a detection stream, and a hypothesized word. An existence feature is present for each unit/word combination seen in the training data, and indicates whether the unit is seen within the hypothesized word's span. Unlike Expectation and Levenshtein features, Existence features do not generalize to new words.

4.4. Baseline and Language Model Features

The language model features were described in Section 3.2. In addition to these, we have developed baseline features that can be used in association with an existing HMM system. For the FDM, we use the language and acoustic model scores of a given hypothesis. For our SCRF system, we have developed an even simpler feature that requires only the baseline one-best sequence, which is treated as a detector sequence.

	FDM	SCRF
Baseline	37.1%	37.1
Existence	36.5	36.7
Expectation	-	36.4
Levenshtein	36.1	36.4

Table 1. Effect of features individually, using multiphone units only. Sentence error rate on complete test set. Expectation features were not implemented in the FDM.

The baseline SCRF feature for a segment is always either $+1$ or -1. It is $+1$ when a hypothesized segment spans exactly one baseline word, and the label of the segment matches the baseline word. Otherwise it is -1. The contribution of the baseline feature to a hypothesis score will be maximized when the hypothesis has the same number of words as the baseline decoding, and the identities of the words match. Thus, by assigning a high enough weight to the baseline feature, the best scoring hypothesis can be guaranteed to be the baseline and thus match its performance. In practice, the baseline weighting is learned and its value will depend on the relative power of the additional features.

5. EXPERIMENTAL RESULTS

While the FDM and SCRF approaches have appeared in the literature before, they have not previously been empirically compared. To do this comparison, we have conducted a series of experiments with data from the Bing Mobile voice-search application [12], which allows users to request local businesses by voice, from their mobile phones. For the purpose of this paper, we set aside 12,758 human-transcribed interactions for evaluation. We further report results on the subset of this test set containing only instances of the 1000 most frequent requests. For parameter tuning, we used a development set of 8,777 utterances. When the full test-set was used, we rescored HMM N-best lists from the baseline system. For training, we used roughly 1.5M spoken queries – 1200 hours of speech – to build an HMM acoustic model which served as our baseline and was also used for generating detector streams. An equal amount of data was used to learn the direct model parameters. Our baseline acoustic model is a conventional ML trained HMM system, using utterance-level mean normalized MFCCs and clustered cross-word triphones. It has 11k context dependent states, and 260k Gaussians. The baseline produces an error rate of 37.1%.

Table 1 compares the FDM and SCRF as multiphone features are added individually to the baseline features. We see that the FDM produces slightly better results, consistent with its ability to create full utterance-level features. Each feature in isolation produces up to 1% gain. Table 2 shows the results when all features are used in both models. We see that there is complementary information resulting in a gain of 1.4-1.8% overall. On the top-1000 business requests, a larger gain of about 2% absolute is observed. Again, the utterance level features of the FDM appear somewhat more effective than the word level features of the SCRF. In terms of runtime, both the SCRF and FDM rescorings - exclusive of the time taken to generate inputs - are much faster than a standard HMM decoding. The bulk of the time is spent generating the phone and multi-phone detections and baseline HMM decoding.

	Baseline	FDM	SCRF
Top-1000	15.8%	13.6	13.9
Full Test	37.1	35.3	35.7

Table 2. Sentence error rates using both phone and multi-phone stream and all features.

6. CONCLUSION

This paper compares two recently developed direct modeling approaches. Both allow for the use of long-span segmental features. The flat direct model operates at the utterance level, and is especially suited to evaluating a small set of candidates. The segmental CRF model operates at the word level and generalizes to continuous speech recognition. Both approaches provide consistent Voice Search improvements.

7. REFERENCES

[1] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in *Proc. EMNLP*, 1996.

[2] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proc. ICML*, 2001.

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

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

[5] J. Morris and E. Fosler-Lussier, "Discriminative Phonetic Recognition with Conditional Random Fields," in *HLT-NAACL*, 2006.

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

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

[8] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *Proc. Machine Learning*, 2000.

[9] 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.

[10] XD. Huang, A. Acero, and H-W. Hon, *Spoken Language Processing*, Prentice Hall, 2001.

[11] G. Zweig and P. Nguyen, "A segmental crf approach to large vocabulary continuous speech recognition," in *Proc. ASRU*, 2009.

[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*, 2007.

[13] S. Sarawagi and W. Cohen, "Semi-Markov Conditional Random Fields for Information Extraction," in *Proc. NIPS*, 2005.

[14] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proc. HLT-NAACL*, 2003.

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