Extensions to the SCARF framework

Patrick Nguyen and Geoffrey Zweig

Microsoft Research Technical Report: MSR-TR-2010-129

September 22, 2010

Contents

Abstract

In this paper, we show simple extensions to the SCARF framework. They are generalization of existing known techniques in machine learning or with Hidden Markov Models which are derived for the particular case of Segmental Conditional Random Fields (SCRF). The extensions are Empircal Bayes Risk training, and Mixture Modeling.

1 Introduction

The SCARF framework as presented [2] offers the basic equations for training SCRFs. We propose to extend the framework with Empirical Bayes Risk training or Empirical Training Error optimization, which are better correlated with the error metric one seeks to optimize, than the conditional log-likelihood normally used. Second, SCRFs are (marginalized) exponential models over segmented sequences. In that model, it is supposed a hyperplane in the feature space is adequate to separate correct words from incorrect ones. We relax this assumption by using mixture models, wherein a weighted sum of exponentials replaces the basic model. Mixture models are useful in various contexts, such as model interpolation for adaptation, and boosting.

These variations are based on a modification of the model functional, or the objective criterion. A simple modification to the gradient equations will achieve our goals. Therefore, we first visit the gradient calculation for the log conditional likelihood. For didactic purposes, we generalize in small incremental steps, from Maximum Entropy (log-linear), Semi-Conditional Random Fields, to SCARF models. Then, we show how to use or modify the gradient functional for the different extensions mentioned above.

1.1 Gradient computations

In this section, we endeavor to expose the calculation of the gradient with respect to the parameter vector for the condtional likelihood. This is intended as a quick tutorial to show step by step how the equations for the gradients appear. First, we show the simple unstructured case called maximum entropy. Second, we generalize this to a segmental model where the segmentation is observed. Finally, we extend this to our SCRF model where segmentation is unobserved and marginalized out.

1.1.1 Maximum Entropy

Simple maximum entropy models operate over two unstructured variables. The observation o, is the input. The output class s, is assumed to be discrete. The functional form of this model is as follows:

$$p(s|o) := \frac{\pi(s|o)}{\sum_{s'} \pi(s'|o)}, \tag{1}$$

where

$$\pi(s|o) := \exp\left[\lambda^T f(s|o)\right]. \tag{2}$$

The feature vector $f(s|o)$ characterizes the input in relation to the output. Because π is exponential in the features, the model is called *exponential* or *log-linear*. In the vernacular of random fields, the function $\pi(\cdot)$ is often called a potential function. The parameter vector λ is what one seeks to alter so as to optimize the log conditional likelihood. For convenience, we call the Z the partition function:

$$Z := \sum_{s'} \pi(s'|o). \tag{3}$$

The objective function is:

$$J := \sum_{t} \log p(s_t|o_t), \tag{4}$$

where each labeled data sampled pair is (o_t, s_t). The objective function is called "linear" because it is a sum of sample-wise objective functions. Therefore, it is sufficient to consider what happens for a single labeled observation (o, s) and we will drop the t index throughout. The objective function, in this case, is convex. Denote the gradient wrt the parameters as $\partial_\lambda := \partial/\partial_\lambda$. The gradient of the objective function is:

$$\begin{aligned}
\partial_\lambda J &= \partial_\lambda \log p(s|o) &(5)\\
&= \frac{1}{p(s|o)} \partial_\lambda p(s|o) &(6)\\
&= \frac{1}{p(s|o)} \left[\frac{\partial_\lambda \pi(s|o)}{Z} - \frac{\pi(s|o)}{Z^2} \partial_\lambda Z \right] &(7)
\end{aligned}$$

remarking that $p(s|o) = \pi(s|o)/Z$:

$$\begin{aligned}
&= \frac{Z}{\pi(s|o)} \frac{1}{Z} \partial_\lambda \pi(s|o) - \frac{1}{p(s|o)} \frac{p(s|o)}{Z} \partial_\lambda Z &(8)\\
&= \frac{1}{\pi(s|o)} \partial_\lambda \pi(s|o) - \frac{1}{Z} \partial_\lambda Z &(9)
\end{aligned}$$

since $\partial_\lambda \pi(s|o) = \pi(s|o) f(s|o)$,

$$= \frac{1}{\pi(s|o)} \pi(s|o) f(s|o) - \frac{1}{Z} \partial_\lambda \sum_{s'} \pi(s'|o) \tag{10}$$

and using the first expression for the gradient of π into the second,

$$\begin{aligned}
&= f(s|o) - \frac{1}{Z} \sum_{s'} \pi(s'|o) f(s'|o) &(11)\\
&= f(s|o) - \sum_{s'} p(s'|o) f(s'|o) = f(s|o) - \mathrm{E}_{S'|o} f(S'|o). &(12)
\end{aligned}$$

It has been noted that a sometimes more convenient form comes in handy:

$$\partial_\lambda J = \sum_{s'} \Big[\delta(s, s') - p(s'|o) \Big] f(s'|o), \tag{13}$$

with $\delta(\cdot, \cdot)$ the Kronecker delta. The last equation shows that the gradient tends to seek correction proportional to the difference between supervision and the probability given by our current model. These are the equations underlying the Flat Direct Model [1].

1.1.2 Semi-Conditional Random Fields

Semi-Conditional Random Fields are, in the same way as maximum entropy models, exponential models. However, both the output and the input are assumed to be structured. Also, the output variable can be partitioned into two parts, one called the output sequence s, and another called the segmentation q, which separates the sequence of input observations o into small batches, later called e.

The functional form of the model is as follows:

$$p(s, q|o) := \frac{\pi(s, q|o)}{\sum_{s',q'} \pi(s', q'|o)}. \tag{14}$$

For more details about the meaning of s and q, please refer to [2]. The Markov assumption underlying this model allows us to break up the π function into a product of exponentials for each segment e, defined entirely independently of each other. That is,

$$\pi(s, q|o) := \prod_{e \in q} G(e|o), \tag{15}$$

with:

$$G(e|o) := \exp(\lambda^T f(e|o)). \tag{16}$$

In this case, the partition function is:

$$Z := \sum_{s',q'} \pi(s', q'|o). \tag{17}$$

Since this is an exponential model, the gradient can be computed exactly as before, with the addition that the factorization of $\pi(s, q|o)$ into small segments e may be accounted for. That is,

$$\begin{aligned}
\partial_\lambda J &= \partial_\lambda \log p(s, q|o) && (18)\\
&= \frac{1}{p(s, q|o)} \partial_\lambda \frac{\pi(s, q|o)}{Z} && (19)\\
&= \frac{1}{p(s, q|o)} \left[\frac{\partial_\lambda \pi(s, q|o)}{Z} - \frac{\pi(s, q|o)}{Z^2} \partial_\lambda Z \right] && (20)\\
&= \frac{Z}{\pi(s, q|o)} \frac{1}{Z} \partial_\lambda \pi(s, q|o) - \frac{\pi(s, q|o)}{p(s, q|o)} \frac{1}{Z^2} \partial_\lambda Z && (21)\\
&= \frac{1}{\pi(s, q|o)} \partial_\lambda \pi(s, q|o) - \frac{1}{Z} \partial_\lambda Z && (22)\\
&= \frac{1}{\pi(s, q|o)} \sum_{e \in q} \prod_{e' \in q, e' \neq e} G(e'|o) \partial_\lambda G(e|o) - \frac{1}{Z} \partial_\lambda Z && (23)
\end{aligned}$$

and, given that $\partial_\lambda G(e|o) = G(e|o) f(e|o)$,

$$\begin{aligned}
&= \frac{1}{\pi(s, q|o)} \sum_{e} \pi(s, q|o) f(e|o) - \frac{1}{Z} \partial_\lambda Z && (24)\\
&= \sum_{e} f(e|o) - \frac{1}{Z} \sum_{s',q'} \pi(s', q'|o) \sum_{e' \in q'} f(e'|o) && (25)\\
&= \sum_{e} f(e|o) - \sum_{s',q'} p(s', q'|o) \sum_{e'} f(e'|o). && (26)
\end{aligned}$$

Refactoring to expose the risk component as previously, we obtain:

$$\partial_\lambda J = \sum_{s',q'} \left[\delta(s,s')\delta(q,q') - p(s',q'|o) \right] \sum_e f(e|o). \tag{27}$$

This is the same as the maximum entropy case, when we set $s \leftarrow (s,q)$.

1.1.3 Segmental Conditional Random Fields (SCRFs)

In speech recognition, it is generally the case that the segmentation q is unobserved during training. Therefore, the probability of the output is partially marginalized over q in order to yield a probability over the ouput token states s. That is,

$$p(s|o) := \sum_q p(s,q|o) = \frac{\sum_q \pi(s,q|o)}{\sum_{s',q'} \pi(s',q'|o)}. \tag{28}$$

The partition function is identical. For convenience, we denote the marginalized potential function:

$$\pi(s|o) := \sum_q \pi(s,q|o). \tag{29}$$

Unlike previous models, the objective function is no longer convex. We follow a similar derivation as in the previous case, with due attention brought on the marginalization over q:

$$\begin{aligned}
\partial_\lambda J &= \partial_\lambda \log p(s|o) && (30)\\
&= \frac{1}{p(s|o)} \left[\partial_\lambda \frac{\pi(s|o)}{Z} \right] && (31)\\
&= \frac{1}{p(s|o)} \left[\frac{\partial_\lambda \pi(s|o)}{Z} - \frac{\pi(s|o)}{Z^2} \partial_\lambda Z \right] && (32)\\
&= \frac{Z}{\pi(s|o)} \frac{1}{Z} \partial_\lambda \pi(s|o) - \frac{1}{p(s|o)} \frac{\pi(s|o)}{Z^2} \partial_\lambda Z && (33)\\
&= \frac{1}{\pi(s|o)} \partial_\lambda \pi(s|o) - \frac{1}{Z} \partial_\lambda Z && (34)\\
&= \frac{1}{\pi(s|o)} \sum_{q,e} \prod_{e' \in q \setminus \{e\}} G(e'|o) \partial_\lambda G(e|o) - \frac{1}{Z} \partial_\lambda Z && (35)\\
&= \frac{1}{\pi(s|o)} \sum_{q,e} \pi(s,q|o) f(e|o) - \frac{1}{Z} \partial_\lambda Z && (36)\\
&= \frac{1}{\pi(s|o)} \sum_{q,e} \pi(s,q|o) f(e|o) - \sum_{s',q'} \frac{\pi(s',q'|o)}{Z} \sum_{e'} f(e'|o) && (37)\\
&= \frac{1}{\pi(s|o)} \sum_{q,e} \pi(s,q|o) f(e|o) - \sum_{s',q'} p(s',q'|o) \sum_{e'} f(e'|o). && (38)
\end{aligned}$$

Rearranging the terms to expose the segment risk as before, we see:

$$\partial_\lambda J = \sum_{s',q'} \left[\frac{\pi(s',q'|o)}{\pi(s'|o)} \delta(s,s') - p(s',q'|o) \right] \sum_e f(e|o) \tag{39}$$

Remarking that:

$$\frac{\pi(s,q|o)}{\pi(s|o)} = \frac{\pi(s,q|o)}{Z} \frac{Z}{\pi(s|o)} = \frac{p(s,q|o)}{p(s|o)} = p(q|o,s),$$

we get:

$$\partial_\lambda J = \sum_{s',q'} \left[p(q'|o,s')\delta(s,s') - p(q'|o,s')p(s'|o) \right] \sum_e f(e|o) \tag{40}$$

$$= \sum_{s',q'} p(q'|o,s') \left[\delta(s,s') - p(s'|o) \right] \sum_e f(e|o). \tag{41}$$

As in the maximum entropy case, we see that the gradient is proportional to the difference between label and probability of our model. In this case, this is true *at the sentence level*. In other words, there is no gradation in seriousness of error, whether one word or all words are wrong in the candidate hypothesis s'. This has been the motivation for much research in the speech recognition community for the case of discriminative training of HMMs. We show here that the same techniques and insight may be applied to our models.

2 Empirical Bayes Risk

In the context of speech recognition, it is understood that sentence level error minimization is inappropriate. In most applications, one wishes to minimize the word-level error rate instead, or a weighted variant thereof. In other words, we will get penalized for each word error individually[1].

To achieve this, it is generally believed that Empirical Bayes Error will help us design a system which attempts to commit fewer word errors, rather than getting sentences exactly correct. The Empirical Bayes Risk is expressed through a function $\mathcal{R}_s(s_x, q_x)$ for a sentence s_x and given a segmentation q_x. For readability, we omit the supervision label s. The Empirical Bayes Risk function is not convex, and defined as:

$$J := \mathrm{E}_{\cdot|o} \mathcal{R}(S_x, Q_x) = \sum_{s_x, q_x \in s_x} p(s_x, q_x|o) \mathcal{R}(s_x, q_x). \tag{42}$$

Furthermore, the risk function as linearly decomposable over the segments (edges of the graph):

$$\mathcal{R}(s_x, q_x) := \sum_{e_x \in q_x} \mathcal{R}(e_x). \tag{43}$$

[1]As a technical detail, the Empirical Bayes Risk criterion is less prone to instability when the numerator constraints are not present in the denominator.

This is true of both the MWE (Minimum Word Error) function and the exact error rate. We are interested in following the gradient wrt the parameter vector:

$$\partial_\lambda J = \partial_\lambda \sum_{s_x, q_x} p(s_x, q_x|o)\mathcal{R}(s_x, q_x) \tag{44}$$

$$= \sum_{s_x, q_x} p(s_x, q_x|o)\mathcal{R}(s_x, q_x)\partial_\lambda \log p(s_x, q_x) \tag{45}$$

Plugging in results from Eq (27):

$$= \sum_{s_x, q_x} p(s_x, q_x|o)\mathcal{R}(s_x, q_x) \sum_{s', q'} \Big[\delta(s_x, s')\delta(q_x, q') - p(s', q'|o)\Big] \sum_{e' \in q'} f(e'|o) \tag{46}$$

$$= \sum_{s', q'} \Big[\mathcal{R}(s', q')p(s', q'|o) - p(s', q'|o)J\Big] \sum_{e'} f(e'|o) \tag{47}$$

$$= \sum_{s', q'} p(s', q'|o) \Big[\mathcal{R}(s', q') - J\Big] \sum_{e'} f(e'|o) \tag{48}$$

$$= \sum_{s', q'} p(s', q'|o) \Big[\sum_{e'_1 \in q'} \mathcal{R}(e'_1) - J\Big] \sum_{e'} f(e'|o). \tag{49}$$

Although it appears that there is a double summation over all edges $\{e \in q'\}$, we can efficiently collect statistics with a forward-backward algorithm similar to that described in [2].

2.1 Forward-backward algorithm

Drawing from the notations in our technical report, we define:

Q_i^j : set of possible segmentations from time i to time j,

S_a^b : set of possible state sequences starting with successor to lattice state a, and ending in lattice state b.

During the forward-backward algorithm, we accumulate the following quantities:

$$\alpha_i(s) := \sum_{s \in S_{<s>}^{s}} \sum_{q \in Q_1^i, |q|=|s|} G(e|o), \tag{50}$$

$$\beta_i(s) := \sum_{s \in S_s^{</s>}} \sum_{q \in Q_{i+1}^N, |q|=|s|} G(e|o). \tag{51}$$

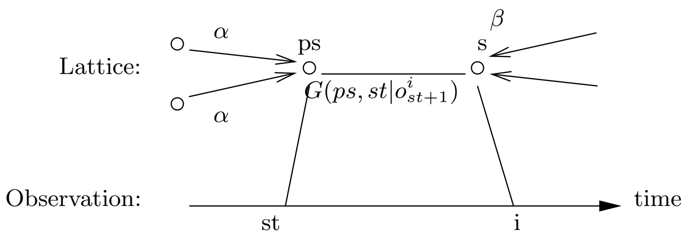

Figure 1: Forward-backward algorithm for an end state s at time i under consideration.

To be clear, we have:

$$\begin{aligned}
N &: \text{total time in the sentence,}\\
i &: \text{a variable referring to current (end) time,}\\
st &: \text{a variable referring to start time,}\\
ps &: \text{a variable referring to a predecessor state, and}\\
s &: \text{a variable referring to the current (end) state.}
\end{aligned}$$

For reference, we represent the algorithm pictorially in Figure 1.

The main purpose of the algorithm is to accumulate efficiently:

$$\gamma_s(st, i) := p(e = [st, i, s]|o) \sum_{ps} \frac{\alpha_{ps}(st)\beta_s(i)}{Z} G(ps, s|o^i_{st+1}). \tag{52}$$

From Eq (41), we can see that we may run a single forward-backward algorithm, and accumulate the additional risk sums:

$$A_s(i) := \sum_{s \in S^s_{<s>}} \sum_{q \in Q^i_1, |q|=|s|} G(e|o)R(e|o), \tag{53}$$

$$B_s(i) := \sum_{s \in S_s^{</s>}} \sum_{q \in Q^N_{i+1}, |q|=|s|} G(e|o)R(e|o). \tag{54}$$

Note that $J = A_{</s>}(N)/Z$, which is available to us at the end of the forward pass. During the backward pass, we accumulate:

$$F(e) := \sum_{st,ps} \left[\frac{A_{ps}(st)B_i(s)}{Z} - \gamma_s(st, i)J \right] f(e|o). \tag{55}$$

2.2 Edge-based risk

Sometimes, an quicker expedient to risk is sought. As an approximation, one might find it more expedient to define a risk function based on edges $(st, i], s$,

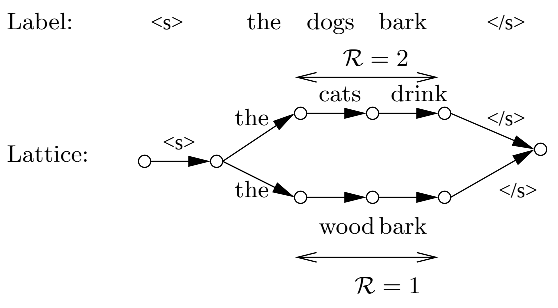

Figure 2: An example of how risk is distributed from paths q to edges.

and sum over the expected posterior. That is,

$$J := \sum_e p(e|o)R(e). \tag{56}$$

The exact approach presented previously, differs with the current in the following ways:

1. The correction to the gradient is given by the difference from the average cost, rather than proportional to the cost.
2. Costs further down in the lattice matter, as shown in the Figure 2. Both are identical iff the lattice is pinched, *i.e.* states and times are identical.

3 Mixture Models

In this section, we define a particular flavor of mixture models and proceed to derive the gradients required for the optimization. The objective function is not convex, and this adds more potential for local optima.

3.1 Sentence mixture models

The first kind of mixture models we envision are sentence-level mixture models. This is comparable to Eigenvoices or Cluster Adaptive Training in HMMs. These mixture models are adequate for sentence-level phenomena, such as topic, domain and gender. We modify the functional form of the model to be:

$$p(s|o) := \frac{\sum_{q,m} w_m \pi_m(s, q|o)}{Z}, \tag{57}$$

and we define:

$$\pi(s|o) := \sum_m \pi_m(s|o), \tag{58}$$

$$\pi_m(s|o) := \sum_q \pi_m(s,q|o), \tag{59}$$

$$\pi_m(s,q|o) := \prod_{e\in q} G_m(e|o), \tag{60}$$

$$G_m(e|o) := \exp[\lambda_m^T f(e|o)], \tag{61}$$

$$Z := \sum_{s',m'} w_{m'}\pi_{m'}(s'|o). \tag{62}$$

Each mixture component has a parameter vector λ_m and a scalar mixture weight w_m. We will restrict mixture weights to be positive, that is $w_m \geq 0$ and $\sum_m w_m > 0$. Without loss of generality, it may be thought that $\sum_m w_m = 1$, in which case it may be thought of as a prior probability.

It is possible to define a mixture of *sets* of features, but for simplicity and wlog we assume a global mixture which applies to all features. We also ignore e, that is, mixtures are independent of the segment. We also ignore the language model state and retain features at the word level only.

Define the gradient operators to be:

$$\partial_\lambda^m := \frac{\partial}{\partial\lambda_m}, \tag{63}$$

$$\partial_w^m := \frac{\partial}{\partial w_m}. \tag{64}$$

The derivation for computing the gradient wrt the mixture component parameter vector λ_m is similar to those explained in the introduction, that is:

$$\begin{aligned}
\partial_\lambda^m J &= \partial_\lambda^m \log p(s|o) && (65)\\
&= \frac{1}{p(s|o)} \partial_\lambda^m \sum_{q \in s, m'} \frac{w_m \pi_{m'}(s, q|o)}{Z} && (66)\\
&= \frac{1}{p(s|o)} \left[\frac{\partial_\lambda^m \pi_m(s|o)}{Z} - \frac{1}{Z^2} \pi(s|o) \partial_\lambda^m Z \right] && (67)\\
&= \frac{Z}{\pi(s|o)} \frac{1}{Z} \partial_\lambda^m \pi(s|o) - \frac{1}{Z} \partial_\lambda^m Z && (68)\\
&= \frac{1}{\pi(s|o)} \sum_{q, e \in q} w_m \pi_m(s, q|o) f(e|o) - \frac{1}{Z} \partial_\lambda^m Z && (69)\\
&= \frac{1}{\pi(s|o)} \sum_{q, e} w_m \pi_m(s, q|o) f(e|o) - \frac{1}{Z} \sum_{s', q', e'} w_m \pi_m(s', q'|o) f(e'|o) && (70)\\
&= \sum_{s', q'} \left[\delta(s, s') \frac{w_m \pi_m(s, q|o)}{\pi(s|o)} - w_m p(s', q'|o, m) \right] \sum_e f(e|o) && (71)\\
&= \sum_{s', q'} p(q|o, s', m) w_m \left[\delta(s, s') - p(s'|o, m) \right] \sum_e f(e|o). && (72)
\end{aligned}$$

As mentioned above, we can normalize weights and correctly interpret:

$$p(q|o, s, m) w_m = p(q, m|o, s). \quad (73)$$

Thus, we have:

$$\partial_\lambda^m \log p(s|o) = \sum_{s', q'} p(q, m|o, s') \left[\delta(s, s') - p(s'|o, m) \right] \sum_e f(e|o). \quad (74)$$

The derivative wrt the mixture weight is:

$$\begin{aligned}
\partial_w^m \log p(s|o) &= \partial_w^m \left[\log \sum_{m'} w_{m'} \pi_{m'}(s|o) - \log Z \right] && (75)\\
&= \frac{1}{\sum_{m'} w_{m'} \pi_m(s|o)} \partial_w^m \sum_{m'} w_{m'} - \frac{1}{Z} \partial_w^m Z && (76)\\
&= \frac{\pi_m(s|o)}{\sum_{m'} w_{m'} \pi_{m'}(s|o)} - \frac{1}{Z} \sum_{s'} \pi_m(s'|o). && (77)
\end{aligned}$$

Remarking that:

$$p(s', m|o) = \frac{w_m \pi_m(s'|o)}{\sum_{s', m'} w_{m'} \pi_{m'}(s'|o)},$$

we obtain:

$$\partial_w^m \log p(s|o) = \frac{1}{w_m} p(s,m|o) - \frac{1}{w_m} \sum_{s'} p(s',m|o) \tag{78}$$

$$= \frac{1}{w_m} \Big[p(s,m|o) - p(m|o) \Big]. \tag{79}$$

The restriction for having positive weights is commonly achieved by optimizing a variable μ_m instead, with

$$w_m := \exp \mu_m. \tag{80}$$

This enforces $w_m > 0$ and thus the weaker requirement that $\sum_m w_m > 0$, since μ_m is required to be finite. In practice, this is implemented with *a prior feature specific to each mixture component weight*.

3.2 Word-level mixtures

Rather than mixtures at the sentence level, it is possible to envision having mixtures at the level of an LM state. Each LM state corresponds to a word. This is comparable, in the HMM framework, to Gaussian mixture models, rather mixture of HMMs. This model has more flexibility in choosing the best model for each word separately. This is useful for word-level phenomena. In practice, we believe it to be more important for overcoming the weakness of the linear model, rather than explicitly modeling physical phenomena, such as gender in the previous case. In a case where each feature is an expert, this model finds the optimal combination between product and sum of experts.

The new functional is:

$$p(s|o) := \frac{\sum_q \prod_e \sum_m w_m G_m(e|o)}{Z}, \tag{81}$$

where

$$G_m(e|o) := \exp \left[\lambda_m^T f(e|o) \right]. \tag{82}$$

Define

$$G(e|o) := \sum_m w_m G_m(e|o). \tag{83}$$

We have:

$$\partial_\lambda^m \log p(s|o) = \partial_\lambda^m \sum_q \prod_e \sum_{m'} w_{m'} G_{m'}(e|o) - \frac{1}{Z}\partial_\lambda^m Z \tag{84}$$

$$= \frac{1}{\pi(s|o)} \sum_{q,e} \left[\prod_{e' \neq e} \sum_{m'} w_{m'} G(e'|o)\right] w_m \partial_\lambda^m G_m(e|o) - \frac{1}{Z}\partial_\lambda^m Z \tag{85}$$

$$= \frac{1}{\pi(s|o)} \sum_{q,e} \left[\prod_{e'} \sum_{m'} w_{m'} G(e'|o)\right] \frac{w_m G_m(e|o)}{\sum_{m'} w_{m'} G_{m'}(e|o)} f(e|o) - \frac{1}{Z}\partial_\lambda^m Z \tag{86}$$

$$= \sum_q \frac{\pi(s,q|o)}{\pi(s|o)} \sum_e \frac{G_m(e|o)}{G(e|o)} f(e|o) - \frac{1}{Z}\partial_\lambda^m Z \tag{87}$$

$$= \sum_{s',q',e'} \frac{G_{m'}(e'|o)}{G(e'|o)} p(q'|o,s') \left[\delta(s,s') - p(s'|o)\right] f(e|o). \tag{88}$$

References

[1] P. Nguyen, G. Heigold, and G. Zweig. Speech Recognition with Flat Direct Models. *IEEE Journal of Selected Topics in Signal Processing*, December 2010.

[2] G. Zweig and P. Nguyen. A Segmental CRF Approach to Large Vocabulary Continuous Speech Recognition. In *ASRU*, 2009.