A Graphical Model for Shallow Parsing Sequences

Adrian Silvescu and Vasant Honavar
Department of Computer Science
Iowa State University, Ames, IA 50010, USA
{silvescu , honavar}@cs.iastate.edu
http://www.cs.iastate.edu/˜silvescu

[**Intro**] Probabilistic Models for sequence data can be mainly divided into two categories: I. Fairly sophisticated models that are aimed at finding an all encompassing characterisation of the whole sequence by the means of an interdependent generative process (such as PCFGs and HMMs); and II. Relatively simple models that make an independence assumption regarding the generation process of each of the elements of the sequence (such as Unigram, Naive Bayes, Clustering, PLSA and LDA).In this paper we explore the interval between these two extremes with a model that does not attempt to be a global characterisation of the sequence as in the case of PCFG/HMM but yet it does not assume independence among the generative processes of the subsequent elements in the sequence.

[**Model**] More exactly we assume that the generation process produces chunks(subsequences) of elements of variable size, which are then concatenated in order to yield the whole sequence. The elements within a chunk are dependent of each other and the dependence is quantified by the hidden macro- variable Gi denoting the current chunk generator. The chunk generators are chosen independently from a distribution whose possible values is the set of all possible chunk generators. [This model can be further extended to dependencies, either directed or undirected among chunk generators but for the sake of simplicity we will stick to the independence assumption in this paper]. The generative process, using plate notation, is depicted in the figure above. Basically, a generator Gi is first drawn from a multinomial distribution and then it is used to generate the subsequence(chunk) Si. The chunk generators in our case are PCFGs(Probabilistic Context Free Grammars), hence the tree model within the hidden variable denoting the generator Gi. The generated sequence is actually a concatenation of independent samples from a mixture of (small!) PCFGs.

[**Inference**] Inference in this model is aimed at determining for each generative model Gi and for each Nonterminal Nj from the PCFGs what is the probability that each of these hidden variables has in generating a subsequence of the output. The resulting algorithm has the flavor of the Inside - Outside algorithm for reasoning in PCFGs but needs to be augmented with one more message aside from the traditional$\alpha$and β, that we call $\eta(u, v)$ (thus yielding the $\alpha\beta\eta$ - algorithm). The additional message $\eta(u, v)$ quantifies the probability that the sequence between u and v is a concatenation of chunks under the current model. A Most Probable Explanation procedure (Viterbi like) can be derived too, by replacing sums with max in the appropriate places in our $\alpha\beta\eta$.

[**Learning - Parameters**] In order to estimate the parameters of the model we use Expectation Maximization. The Expectation step computes the probabilities outlined in the previous paragraph using the $\alpha\beta\eta$ - algorithm. As for Maximization, since our model is decomposable - it has sufficient statistics. Therefore all we have to do in order to compute the maximum likelihood estimator for the parameters is to compute the expected counts by summing up the probabilities obtained during the Expectation step. [Some regularization is actually also going on in the form or Dirichlet priors but for simplicity ...]

[**Note**] Note that even though the $\alpha\beta\eta$ algorithm is conceptually more complicated that the $\alpha\beta$ we obtain considerable improvements in running time because the $\alpha\beta$ in our case is run on much smaller strings (chunks vs whole seq.)

[**Learning - Structure**] In order to learn the structure of the PCFGs we use a Structural-EM type of approach with appropriate custom operators designed for this task. We can show that every CFG can be written in and Abstraction Super-structuring Normal Form (ASNF), which restricts the productions to one of the following types: 1.[Atomic] $A \rightarrow a$ and this is the only production that contains A in the LHS; 2. [Super-structuring] $A \rightarrow BC$ and this is the only production that contains A in the LHS; and 3. [Abstraction] $A \rightarrow B$. As a consequence our Structural - EM operators will be of either the Abstraction or Super-structuring type and with proper randomization we can show that the procedure is complete. All the Atomics are inserted in the first step by default.

[**Experiments**] The experiments are targeted at evaluating the performance of the proposed approach on text and protein sequences tasks.