Exploiting Syntactic, Semantic and Lexical Regularities in Language Modeling via Directed Markov Random Fields

Shaojun Wang[‡] SWANG@CS.UALBERTA.CA
Shaomin Wang[†] SMWANG@MIT.EDU
Russell Greiner[‡] GREINER@CS.UALBERTA.CA
Dale Schuurmans[‡] DALE@CS.UALBERTA.CA
Li Cheng[‡] LICHENG@CS.UALBERTA.CA
[‡]University of Alberta
[†]Massachusetts Institute of Technology

Abstract

We present a directed Markov random field (MRF) model that combines n-gram models, probabilistic context free grammars (PCFGs) and probabilistic latent semantic analysis (PLSA) for the purpose of statistical language modeling. Even though the composite directed MRF model potentially has an exponential number of loops and becomes a context sensitive grammar, we are nevertheless able to estimate its parameters in cubic time using an efficient modified EM method, *the generalized inside-outside algorithm*, which extends the inside-outside algorithm to incorporate the effects of the n-gram and PLSA language models. We generalize various smoothing techniques to alleviate the sparseness of n-gram counts in cases where there are hidden variables. We also derive an analogous algorithm to calculate the probability of initial subsequence of a sentence, generated by the composite language model. Our experimental results on the Wall Street Journal corpus show that we obtain significant reductions in perplexity compared to the state-of-the-art baseline trigram model with Good-Turing and Kneser-Ney smoothings.

1. Introduction

The goal of statistical language modeling is to accurately model the probability of naturally occurring word sequences in human natural language. The dominant motivation for language modeling has traditionally come from the field of speech recognition (Jelinek 1998), however statistical language models have recently become more widely used in many other application areas, such as information retrieval, machine translation and bioinformatics.

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005.

There are various kinds of language models that can be used to capture different aspects of natural language regularity. The simplest and most successful language models are the Markov chain (n-gram) source models, first explored by Shannon in his seminal paper (Shannon 1948). These simple models are effective at capturing local lexical regularities in text. However, many recent approaches have been proposed to capture and exploit different aspects of natural language regularity, sentence-level syntactic structure (Chelba and Jelinek 2000, Roark 2001) and document-level semantic content (Bellegarda 2000, Hofmann 2001), with the goal of outperforming the simple n-gram model. Unfortunately each of these language models only targets some specific, distinct linguistic phenomena. The key question we are investigating is how to model natural language in a way that simultaneously accounts for the lexical information inherent in a Markov chain model, the hierarchical syntactic structure captured in a stochastic branching process, and the semantic content embodied by a bag-of-words mixture of log-linear models—all in a unified probabilistic framework.

Several techniques for combining language models have been investigated. The most commonly used method is simple linear interpolation (Chelba and Jelinek 2000, Rosenfeld 1996), where each individual model is trained separately and then combined by a weighted linear combination. The weights in this case are trained using held out data. Even though this technique is simple and easy to implement, it does not generally yield effective combinations because the linear additive form is too blunt to capture subtleties in each of the component models. Another approach is based on Jaynes' maximum entropy (ME) principle (Berger et al. 1996, Khudanpur and Wu 2000, Rosenfeld 1996) which was first applied in language modeling a decade ago, and has since become a dominant technique in statistical natural language processing. It is now well known that for complete data, the ME principle is equivalent to maximum likelihood estimation (MLE) in an undirected Markov random field. In fact, these two problems are exact duals of one another (Berger, et al. 1996). The major weakness with ME methods, however, is that they can only model distributions over explicitly observed features, whereas in natural language we encounter hidden semantic (Bellegarda 2000, Hofmann 2001) and syntactic information (Chelba and Jelinek 2000). Recently Wang et al. (2003) proposed the latent maximum entropy (LME) principle, which extends standard ME estimation by incorporating hidden dependency structure. However, when they apply LME to build a composite language model, they have been unable to incorporate PCFGs in this framework, because the tree-structured random field component creates intractability in calculating the feature expectations and global normalization over an infinitely large configuration space. Previously they had envisioned that MCMC sampling methods (Wang et al. 2005) would have to be employed, leading to enormous computational expense.

In this paper, instead of using an undirected MRF model, we present a unified generative *directed Markov random field model* framework that combines n-gram models, PCFG and PLSA. Unlike undirected MRF models where there is a global normalization factor over an infinitely large configuration space, which often causes computational difficulty, the directed MRF model representation for the composite n-gram/syntactic/semantic model only requires many local normalization constraints. More importantly it satisfies certain factorization property which greatly reduces the computational burden and makes the optimization tractable. We learn the composite model by exploiting the factorization properties of the composite model, so we can use a simple yet efficient EM iterative optimization method, *the generalized inside-outside algorithm*, which enhances the well known inside-outside algorithm (Baker 1979) to incorporate the effects of the n-gram and PLSA language models. Given that n-gram, PCFG and PLSA models have each been well studied, it is striking that this procedure has gone undiscovered until now.

2. A Composite Trigram/Syntactic/Semantic Language Model

Natural language encodes messages via complex, hierarchically organized sequences. The local lexical structure of the sequence conveys surface information, while the syntactic structure, encoding long range dependencies, carries deeper semantic information.

Let X denote a set of random variables $(X_\tau)_{\tau\in\Gamma}$ taking values in a (discrete) probability spaces $(\mathcal{X}_\tau)_{\tau\in\Gamma}$ where Γ is a finite set of states. We define a (discrete) directed Markov random field to be a probability distribution $\mathcal{P}$ which admits a recursive factorization if there exist nonnegative functions, $k^\tau(\cdot,\cdot), \tau\in\Gamma$ defined on $\mathcal{X}_\tau\times\mathcal{X}_{pa(\tau)}$, such that $\sum_{x_\tau} k^\tau(x_\tau, x_{pa(\tau)}) = 1$ and $\mathcal{P}$ has density

$$p(x) = \prod_{\tau\in\Gamma} k^\tau(x_\tau, x_{pa(\tau)}) \tag{1}$$

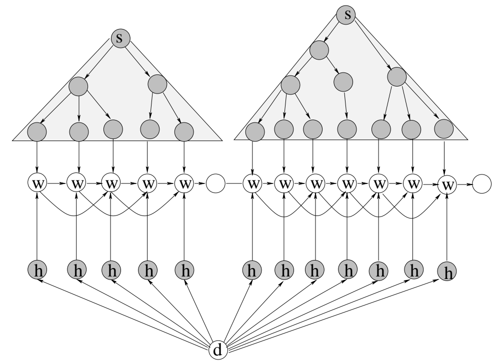


Figure 1. The observables in natural language consist of words, sentences, and documents; whereas the hidden data consists of sentence-level syntactic structure and document-level semantic content. The figure illustrates a composite chain/tree/table model incorporating these aspects, where light nodes denote observed information and dark nodes/triangles denote hidden information.

If the recursive factorization respects to a graph $\mathcal{G}$, then we have a Bayesian network (Lauritzen 1996). But broadly speaking, the recursive factorization can respect to a more complicated representation other than a graph which has a fixed set of nodes and edges.

Assume that we use a trigram Markov chain to model local lexical information, a PCFG to model the syntactic structure and a PLSA (Pritchard et al. 2000, Hofmann 2001) to model its semantic content of natural language, see Figure 1. Each of these models can be represented as a directed MRF model. If we combine these three models, we obtain a composite model that is represented by a rather complex chain-tree-table directed MRF model.

A context free grammar (CFG) (Baker 1979) G is a 4-tuple $(\Sigma, \mathcal{V}, \mathcal{R}, S)$ that consists of: a set of non-terminal symbols Σ whose elements are grammatical phrase markers; a vocabulary of $\mathcal{V} = \{v_1, \cdots, v_M\}$ whose elements, words v_i, are terminal symbols of the language; a sentence "start" symbol $S \in \Sigma$; and a set of grammatical production rules $\mathcal{R}$ of the form: $A \to \gamma$, where $A \in \Sigma$ and $\gamma \in (\Sigma \cup \mathcal{V})^*$. A PCFG is a CFG with a probability assigned to each rule, such that the probabilities of all rules expanding a given nonterminal sum to one. A PCFG is a branching process and can be treated as a directed MRF model, although the straightforward representation as a complex directed graphical model is problematic.

A PLSA (Hofmann 2001) is a generative model of word-document co-occurrences using the bag-of-words assumption as follows: choose a document d with probability $\theta(d)$, select a semantic class h with probability $\theta(d \to h)$, pick a word w with probability $\theta(h \to w)$. The joint probability model for pair of (d, w) is a mixture of log-linear model with the expression $p(d, w) = \theta(d)\sum_h \theta(h \to w)\theta(d \to h)$. The latent class variables function as bottleneck variables to constrain word occurrences in documents.

When a PCFG is combined with a trigram model and PLSA, the grammar becomes context sensitive. If we view each uvw trigram as $uv \to w$, where $u, v, w \in \mathcal{V}$, then the composite trigram/syntactic/semantic language model can be represented as a directed MRF model, where the generation of nonterminals remains the same as in PCFG, but the generation of each terminal depends additionally on its surrounding context; i.e., not only its parent nonterminal but also the preceding two words as well as its semantic content node h.

3. Training Algorithm for the Composite Model

We are interested in learning a composite trigram/syntactic/semantic model from data. We assume we are given a training corpus $\mathcal{W}$ consisting of a collection of documents $\mathcal{D}$, where each document contains a collection of sentences, and each sentence W is composed of a sequence of words from a vocabulary $\mathcal{V}$. For simplicity, but without loss of generality, we assume that the PCFG component of the composite model is in Chomsky normal form. That is, each rule is either of the form $A \to BC$ or $A \to w$ where $B, C \in \Sigma, w \in \mathcal{V}$. When combined with trigram and PLSA models, the terminal production rule $A \to w$ becomes $uvAh \to w$. By examining Figure 1, it should be clear that the likelihood of the observed data under this composite model can be written as below:

$$L(\mathcal{W}, \theta) = \prod_{d \in \mathcal{D}} \left(\prod_l \left(\sum_{H_l} \left(\sum_t p_\theta(d, W_l, H_l, t) \right) \right) \right) \qquad (2)$$

where

$$p_\theta(d, W_l, H_l, t) = \prod_{d \in \mathcal{D}} \left(\prod_l \left(\prod_{h \in \mathcal{H}} \theta(d \to h)^{n(d, W_l, h)} \prod_{u,v \in \mathcal{V}, A \to w \in \mathcal{R}, h \in \mathcal{H}} \theta(uvAh \to w)^{n(uvAh \to w; d, W_l, t, h)} \prod_{A \to BC \in \mathcal{R}} \theta(A \to BC)^{n(A \to BC; d, W_l, t)} \right) \right)$$

here $p_\theta(d, W_l, H_l, t)$ is the probability of generating sentence W_l in document d with parse tree t and semantic content sequence H_l, $n(d, W_l, h)$ is the count of semantic content h in sentence W_l of the document d, $n(uvAh \to w; d, W_l, t, h)$ is the count of trigrams uvw, the non-terminal symbol A and semantic content h in sentence W_l of document d with parse tree t and $n(A \to BC; d, W_l, t)$ is the count of nonterminal production rule $A \to BC$ in sentence W_l of document d with parse tree t. The parameters $\theta(d \to h), \theta(uvAh \to w), \theta(A \to BC)$ are *locally normalized* so that $\sum_{w \in \mathcal{V}} \theta(uvAh \to w) = 1, \sum_{BC \in \Sigma} \theta(A \to BC) = 1, \sum_{h \in \mathcal{H}} \theta(d \to h) = 1$. Thus we have a constrained optimization problem, and there will be a Lagrange multiplier for $uvAh$, nonterminal A and document d.

3.1. Estimating Parameters of the Composite Model

At a first glance, it seems that estimating parameters of the composite model is intractable since the composite directed MRF model potentially has an exponential number of loops, which suggests that loopy belief propagation (Yedidia et al. 2001) and/or variational approximation methods (Wainwright and Jordan 2003) have to be used. It turns out that this is not the case and there is an efficient and exact recursive EM iterative optimization procedure to perform this task.

Following Lafferty's (2000) derivation of the inside-outside formulas for updating the PCFG parameters from a general EM (Dempster et al., 1977) algorithm, we derive the generalized inside-outside algorithm for the composite language model. To apply the EM algorithm, we consider the auxiliary function

$$Q(\theta', \theta) = \sum_d \sum_l \sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) \log \frac{p_{\theta'}(d, W_l, H_l, t)}{p_\theta(d, W_l, H_l, t)}$$

Because of the local normalization constraints, the reestimated parameters of the composite model are then the normalized conditional expected counts:

$$\begin{aligned}
&\theta'(A \to BC) \\
&= \frac{\sum_{d \in \mathcal{D}} \sum_l \sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(A \to BC; d, W_l, t)}{\text{normalization over } BC} \\
&\theta'(uvAh \to w) \\
&= \frac{\sum_{d \in \mathcal{D}} \sum_l \sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(uvAh \to w; d, W_l, t, h)}{\text{normalization over } w} \\
&\theta'(d \to h) \\
&= \frac{\sum_l \sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(d \to h; d, W_l, h)}{\text{normalization over } h}
\end{aligned} \qquad (3)$$

This looks very similar as the PCFG model. Thus we need to compute the conditional expected counts:

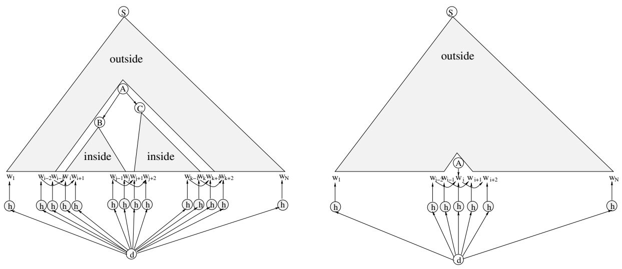
$$\sum_{d \in \mathcal{D}} \sum_l \sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(A \to BC; d, W_l, t)$$

$$\sum_{d \in \mathcal{D}} \sum_l \sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(uvAh \to w; d, W_l, t, h)$$

$$\sum_l \sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(d \to h; d, W_l, h)$$

In general, the sum requires summing over an exponential number of parse trees. However, just as with standard PCFGs, it is easy to check that the following equations still hold

$$\begin{aligned}
&\sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(A \to BC; d, W_l, t) \\
&= \frac{\theta(A \to BC)}{p_\theta(d, W_l)} \frac{\partial p_\theta(d, W_l)}{\partial \theta(A \to BC)} \\
&\sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(uvAh \to w; d, W_l, t, h) \\
&= \frac{\theta(uvAh \to w)}{p_\theta(d, W_l)} \frac{\partial p_\theta(d, W_l)}{\partial \theta(uvAh \to w)} \\
&\sum_{H_l} \sum_t p_\theta(H_l, t | d, W_l) n(d \to h; d, W_l, h) \\
&= \frac{\theta(d \to h)}{p_\theta(d, W_l)} \frac{\partial p_\theta(d, W_l)}{\partial \theta(d \to h)}
\end{aligned}$$



(a) (b)

Figure 2. Inside and outside probabilities in the composite trigram/syntactic/semantic model, where each component is influenced by the injected trigram and PLSA nodes.

and it turns out that there is an efficient way of computing the partial derivative on the righthand side, *the generalized inside-outside algorithm*.

Let $A \Rightarrow \gamma$ denote that, beginning with a nonterminal A, we can derive a string γ of words and nonterminals by applying a sequence of rewrite rules from the grammar *with the flowing-in trigrams and PLSA nodes*, where flowing-in trigrams and PLSA nodes are those that induce the words of the string γ.

Suppose the position of a rule $A \to BC$ within a tree t for sentence $W_l = (w_1, ..., w_N)$ in document d can be specified by a triple $(i, j, k), i \leq j \leq k$. The partial derivative of the probability $p_\theta(S \to W_l \text{ in } d) = p_\theta(d, W_l)$ with respect to the parameter $\theta(A \to BC)$ only involves those parse trees which use the rule $A \to BC$. Consider the event "$S \to W_l$ in d using $A \to BC$ in position (i, j, k)". Because of the Markov property of the directed MRF model, the probability of this event can be written as a product of four terms, i.e., *the factorization property*, as follows:

$$\begin{aligned} & p_\theta(S \to W_l \text{ in } d; \text{ using } A \to BC \text{ in position } (i,j,k)) \\ = \quad & \theta(A \to BC) p_\theta(B \Rightarrow w_i \cdots w_j; W_l \text{ in } d) \\ & p_\theta(C \Rightarrow w_{j+1} \cdots w_k; W_l \text{ in } d) \\ & p_\theta(S \Rightarrow w_1 \cdots w_{i-1} A w_{k+1} \cdots w_N; W_l \text{ in } d) \end{aligned}$$

See Figure 2 (a) for an illustration. The *key insight* toward a solution for the composite model is that, in comparison with the PCFG model, there are additional trigrams which connect the decomposition in position (i, j, k). These dependencies encode additional information from the trigram model, and significantly influence the parameter estimation of the non-terminal grammatical production rules (the impact of the PLSA model is implicitly considered, this will become clear when we derive the estimation formula for the terminal grammatical production rules). The factorization property is the crucial constituent for the success to derive an efficient and exact recursive algorithm.
From this it is not difficult to see that

$$\begin{aligned} & \frac{\partial p_\theta(S \to W_l \text{ in } d)}{\partial \theta(A \to BC)} \\ = & \sum_{i \leq j \leq k} p_\theta(B \Rightarrow w_i \cdots w_j; W_l \text{ in } d) p_\theta(C \Rightarrow w_{j+1} \cdots w_k; W_l \text{ in } d) \\ & \quad p_\theta(S \Rightarrow w_1 \cdots w_{i-1} A w_{k+1} \cdots w_N; W_l \text{ in } d) \end{aligned}$$

Thus, the conditional expected number of times that the rule $A \to BC$ is used in generating the sentence $W_l \in \mathcal{W}$ in document d using the model θ is given by

$$\begin{aligned} & \sum_{H_l} \sum_t p_\theta(t|d, W_l) n(A \to BC; d, W_l, t) = \frac{\theta(A \to BC)}{p_\theta(W_l \text{ in } d)} \\ & \Big(\sum_{i \leq j \leq k} \beta_{ik}(A; W_l \text{ in } d) \alpha_{ij}(B; W_l \text{ in } d) \alpha_{j+1k}(C; W_l \text{ in } d) \Big) \end{aligned}$$

where $\alpha_{ij}(A; W_l \text{ in } d) = p_\theta(A \Rightarrow w_i \cdots w_j; W_l \text{ in } d)$

i.e., the inside probability that the nonterminal A, trigram parent nodes of w_i, w_{i+1} and document node d derive the word subsequence $w_i \cdots w_j$ in the sentence W_l of document d; and
$\beta_{ik}(A; W_l \text{ in } d) = p_\theta(S \Rightarrow w_1 \cdots w_{i-1} A w_{k+1} \cdots w_N; W_l \text{ in } d)$

i.e., the outside probability that beginning with the start symbol S, trigram parent nodes of w_{k+1}, w_{k+2} and document node d, we can derive the sequence $w_1 \cdots w_{i-1} A w_{k+1} \cdots w_N$ in the sentence W_l of document d.

Similarly consider the event "$S \to W_l$ using $uvAh \to w$ in d in position (i)". Because of the Markov property of the directed MRF model, the probability of this event can be written as a product of four terms, again *the factorization property*, as follows:

$$\begin{aligned} & p_\theta(S \to W_l \text{ in } d; \text{ using } uvAh \to w \text{ in position } (i)) \\ & = \delta_{uvw}(w_{i-2} w_{i-1} w_i) \Big(\theta(d \to h) \theta(uvAh \to w) \Big) \\ & \quad p_\theta(S \Rightarrow w_1 \cdots w_{i-1} A w_{i+1} \cdots w_N; W_l \text{ in } d) \end{aligned}$$

See Figure 2 (b) for illustration. The *key insight* toward a solution for the composite model is that comparing with the PCFG model, there are additional trigram and PLSA nodes which connect the decomposition in position (i) to encode the information of both trigram and PLSA nodes and make influencial impact for parameter estimation of the grammatical production rules $uvAh \to w$. Again, the factorization property is the crucial constituent for the success to derive an efficient and exact recursive algorithm.

Thus we have

$$\begin{aligned} & \sum_{H_l} \sum_t p_\theta(H_l, t|d, W_l) n(uvAh \to w; d, W_l, t) = \frac{\theta(uvAh \to w)}{p_\theta(W_l \text{ in } d)} \\ & \sum_{1 \leq i \leq N} \delta_{uvhw}(w_{i-2} w_{i-1} h w_i) \theta(d \to h) \beta_{ii}(A; W_l \text{ in } d) \end{aligned}$$

where δ is the indicator function.

Now consider the event "$S \to W_l$ in d using $d \to h$ in position (i)". Because of the Markov property of the directed MRF model, the probability of this event can be written as sums of products of three terms as follows:

$$\begin{aligned} & p_\theta(S \to W_l \text{ in } d; \text{ using } d \to h \text{ in position } (i)) \\ & = \sum_{A \in \Sigma} p_\theta(S \Rightarrow w_1 \cdots w_{i-1} A w_{i+1} \cdots w_N; W_l \text{ in } d) \\ & \quad \Big(\theta(d \to h) \theta(w_{i-2} w_{i-1} A h \to w_i) \Big) \end{aligned}$$

Thus we have

$$\sum_{H_l}\sum_t p_\theta(H_l|d,W_l)n(d\to h;d,W_l)=\frac{\theta(d\to h)}{p_\theta(W_l\text{ in }d)}\sum_{1\le i\le N}\sum_{A\in\Sigma}\theta(w_{i-2}w_{i-1}Ah\to w_i)\beta_{ii}(A;W_l\text{ in }d)$$

Just as in the PCFG case, there is an efficient recursive method for computing the α's and β's using the CYK chart-parsing algorithm (Young 1967). The only modification is to the definition of α_{ii} so that it incorporates additional information from the trigrams and PLSA nodes. The method for doing this is almost the same as for PCFG and is implicit in the following recursive formulas:

$$\alpha_{ij}(A;W_l\text{ in }d)=\sum_{BC}\sum_{i\le k\le j}\theta(A\to BC)\alpha_{ik}(A;W_l\text{ in }d)\alpha_{k+1j}(C;W_l\text{ in }d)$$

$$\alpha_{ii}(A;W_l\text{ in }d)=\sum_h\theta(d\to h)\theta(w_{i-2}w_{i-1}Ah\to w_i)$$

$$\beta_{ij}(A;W_l\text{ in }d)=\sum_{B,C}\sum_{k<i}\theta(B\to CA)\alpha_{ki-1}(C;W_l\text{ in }d)\beta_{kj}(B;W_l\text{ in }d)+\sum_{B,C}\sum_{k>j}\theta(B\to AC)\alpha_{j+1k}(C;W_l\text{ in }d)\beta_{ik}(B;W_l\text{ in }d)$$

$$\beta_{1N}(A;W_l\text{ in }d)=\delta_S(A;W_l\text{ in }d)$$

Chi (1999) proved that the maximum likelihood estimate of production rule probabilities for a PCFG yields a *proper distribution*, i.e., there is no probability mass lost to infinitely large trees. Similarly we can show that the maximum likelihood estimate of production rule probabilities for this composite model always yields a proper distribution. Due to space limitation, we omit the proof here.

Theorem 1 *Let Ω be the set of finite parse trees, $\hat{p}$ be any intermediate iteration of the EM procedure within the generalized inside-outside algorithm. Then $\hat{p}(\Omega)=1$.*

3.2. Smoothing Techniques of the Composite Model

Current smoothing techniques only handle explicit counts, but in our case there are hidden variables A and h in parameter estimation formula for $\theta(uvAh\to w)$. In this section, we show how to extend smoothing methods to situations where there exist hidden variables.

Notice that the sparse data problem arises from trigram counts. The Good-Turing estimate (Chen and Goodman 1999) is central to combat this problem. The Good-Turing estimate states that for any trigram that occurs n times, we should pretend that it occurs n^* times where

$$n^*=(n+1)\frac{r_{n+1}}{r_n}\tag{4}$$

where r_n is the number of trigrams that occur exactly n times in the training data. To convert this count to a probability, we just normalize: for a trigram vuw with n counts, we take

$$P_{GT}(uvw)=\frac{n^*}{N}\tag{5}$$

where $N=\sum_{n=0}^{\infty}r_n n^*$. In practice, the Good-Turing estimate is not used alone, instead it is often enhanced with back-off technique to combine higher-order models with lower-order models necessary for good performance.

A procedure of replacing a count n with a modified count n^* is called "discount" and we define the ratio $\rho_n=\frac{n^*}{n}$ as a discount coefficient. The ρ_n are calculated as follows: large counts are taken to be reliable, so they are not discounted. In particular, Katz (1987) takes $\rho_n=1$ for all $n\ge k$ for some k. The discount ratios for the lower counts $n\le k$ are derived from the Good-Turing estimate applied to the global trigram distribution and is given as

$$\rho_n=\frac{\frac{n^*}{n}-\frac{(k+1)r_{k+1}}{r_1}}{1-\frac{(k+1)r_{k+1}}{r_1}}\tag{6}$$

When we use (3) to estimate $\theta(uvAh\to w)$, we use the expected count of $n(uvAh\to w)$ where A and h are hidden. However, when the trigram uvw has count $n(uv\to w)>0$, if we discount the expected count of $n(uvAh\to w)$ by the ratio $\rho_n(uvw)$, then we discount the trigrams by the same ratio $\rho_n(uvw)$ since $\sum_{A\in\Sigma,h\in\mathcal{H}}\rho_n(uvw)n(uvAh\to w)=\rho_n(uvw)n(uv\to w)$. Therefore instead of using iterative parameter estimation of (3), we use smoothed iterative parameter estimation as the following,

$$\theta_s'(uvAh\to w)=\frac{\rho_n(uvw)\sum_{d\in\mathcal{D}}\sum_l\sum_{H_l}\sum_t p_\theta(H_l,t|d,W_l)n(uvAh\to w;d,W_l,t,h)}{\text{normalization over }w}$$

When the trigram uvw has count $n(uv\to w)=0$, we backoff to the corresponding bigram parameters and let

$$\theta_s(uvAh\to w)=\eta(uvw)\cdot\theta_s(vAh\to w)$$

and

$$\eta(uvw)=\frac{1-\sum_{w:n(uvw)>0}\theta_s(uvAh\to w)}{1-\sum_{w:n(uvw)>0}\theta_s(vAh\to w)}$$

Similarly we can use Kneser-Ney smoothing technique. Due to space limitation, we omit the details here.

4. Computing the Probability of Initial Subsequence Generation

In automatic speech recognition or statistical machine translation, we are presented with words one at a time, in sequence. Therefore, we would like to calculate the probability $p_\theta(S\to w_1w_2\cdots w_k\cdots)$; that is, the probability that an arbitrary word sequence $w_1w_2\cdots w_k$ is the initial subsequence of a sentence generated by the composite trigram/syntactic/semantic language model. We derive the *generalized left-to-right inside* algorithm to perform this computation by following the work of (Jelinek and Lafferty, 1991), which assumes that a PCFG model is used.

Let $p_\theta(A \ll i, j)$ denote the sum of the probabilities of all trees with root node A and document node d resulting in word sequences whose initial subsequence is $w_i \cdots w_j$. Thus

$$\begin{aligned} p_\theta(A{\ll}i,j) &= \alpha_{ij}(A) + \sum_{x_1\in\mathcal{V}} p_\theta(A \to w_i \cdots w_j x_1) \\ &+ \sum_{x_1x_2\in\mathcal{V}^2} p_\theta(A \to w_i \cdots w_j x_1 x_2) + \cdots \\ &+ \sum_{x_1\cdots x_n\in\mathcal{V}^n} p_\theta(A \to w_i \cdots w_j x_1 \cdots x_n) + \cdots \end{aligned}$$

Using this notation, the desired probability $p_\theta(S \to w_1 w_2 \cdots w_k \cdots)$ is denoted by $p_\theta(S{\ll}1, k)$.

Let $p^L_\theta(A \to B) = \sum_{B_2\in\Sigma} p^L_\theta(A \to B_1 B_2)$ be the sum of the probabilities of all the rules $A \to B_1 B_2$ whose first lefthand side element is $B_1 = B$. Define $p^L_\theta(A \Rightarrow B) = \sum_{\gamma\in(\Sigma\cup\mathcal{V})^*} p_\theta(A \Rightarrow B\gamma)$ as the sum of probabilities of all trees with root node B that produce A as the leftmost first nonterminal. This term converges, since our underlying composite language model p_θ is proper.

Using this definition, we get

$$p_\theta(A << i, i) = \alpha_{i,i}(A) + \sum_{B\in\Sigma} p^L_\theta(A \Rightarrow B)\alpha_{i,i}(B)$$

Define the sum of probabilities of all trees with root node A whose last leftmost production results in leaves B_1 and B_2 as

$$\begin{aligned} p^L_\theta(A \Rightarrow B_1 B_2) &= p_\theta(A \to B_1 B_2) \\ &+ \sum_{C\in\Sigma} p^L_\theta(A \Rightarrow C) p_\theta(C \to B_1 B_2) \end{aligned} \tag{7}$$

Obviously,

$$\begin{aligned} p_\theta(A{\ll}i, i{+}n) = &\sum_{B,B_2\in\Sigma} p^L_\theta(A{\to}B_1B_2) \\ &\Big(\alpha_{i,i}(B_1) p_\theta(B_2{\ll}i{+}1, i{+}n) \\ &+\alpha_{i,i+1}(B_1) p_\theta(B_2{\ll}i{+}2, i{+}n) + \cdots \\ &+\alpha_{i,i+n-1}(B_1 \Rightarrow w_i \cdots w_{i+n-1}) p_\theta(B_2{\ll}i{+}n, i{+}n) \\ &+p_\theta(B_1{\ll}i, i{+}n)\Big) \end{aligned} \tag{8}$$

since to generate the initial subsequence $w_i w_{i+1} \cdots w_{i+n}$, some rule $A \to B_1 B_2$ must first be applied and then the first part of the subsequence must be generated from B_1 and its remaining part from B_2.

Define the sums in the bracket of (8) except the last term as $R(B_1, B_2)$. Then we have

$$\begin{aligned} &p_\theta(A{\ll}i, i{+}n) \\ = &\sum_{B,B_2\in\Sigma} p^L_\theta(A{\Rightarrow}B_1B_2) R(B_1, B_2) \\ &+ \sum_{B_1\in\Sigma} p^L_\theta(A \to B_1) p(B_1{\ll}i, i{+}n) \\ = &\cdots\cdots \\ = &\sum_{B,B_2\in\Sigma} p^L_\theta(A{\Rightarrow}B_1B_2) R(B_1, B_2) \\ &+ \sum_{C_1,\cdots,C_k\in\Sigma, k\to\infty} \Big(p^L_\theta(A{\to}C_1) \prod_{l=2}^{k} p^L_\theta(C_{l-1}{\to}C_l) \\ &p(C_k{\ll}i, i{+}n)\Big) \end{aligned}$$

We have shown that the maximum likelihood estimate of the composite language yields a proper distribution in Theorem 1, thus the last term of the above equation tends to 0 as k grows without limit. Then using definition (7) and successive resubstitutions, we get the final formula

$$\begin{aligned} &p_\theta(A{\ll}i, i{+}n) \\ = &\sum_{B,B_2\in\Sigma} p^L_\theta(A{\Rightarrow}B_1B_2) R(B_1, B_2) \\ = &\sum_{B,B_2\in\Sigma} p^L_\theta(A{\Rightarrow}B_1B_2) \\ &\left(\sum_{j=1}^{n} \alpha_{i,i+j-1}(B_1) p_\theta(B_2{\ll}i{+}j, i+n)\right) \end{aligned}$$

Comparing with a PCFG, the only difference is the way that $R(B_1, B_2)$ is recursively calculated by α, which here takes into account the impact of the trigram and PLSA models.

5. Experimental Evaluation

5.1. Experimental data sets

The corpus used to train our model was taken from the WSJ portion of the NAB corpus, which was composed of about 150,000 documents spanning the years 1987 to 1989, comprising approximately 42 millions words. The vocabulary was constructed by taking the 20,000 most frequent words of the training data. The PCFG production rules we use are extracted from the sections 2-21 of the WSJ treebank corpus. The test set consists of 153,000 words taken from the year 1989.

5.2. Computation in Testing

Since the representation for a document of the test data is not contained in the original training corpus, we use "fold-in" heuristic approach similar to the one used in (Hofmann 2001): the parameters corresponding to the document-semantic arcs, $\theta(d \to h)$, are re-estimated by maximizing the probability of word subsequence currently seen, $w_1, \cdots, w_k$, i.e., the initial subsequence of a sentence generated by the composite language model, while holding the other parameters fixed.

In this case, we use the recursive gradient ascent to update $\theta(d \to h)$.

$$\theta(d \to h)^{(k)} = \theta(d \to h)^{(k-1)} - \frac{\partial \log p_\theta(S << 1, k)}{\partial\theta(d \to h)}\Big|_{\theta^{(k-1)}}$$

We can then recursively calculate the gradient of log-likelihood of the initial subsequence of a sentence with respect to the parameters of document-semantic arc.

5.3. Experimental design

To serve as a baseline standard of performance, we use a conventional trigram model with Good-Turing back-off and Kneser-Ney smoothing. Implementing these approaches, we obtained perplexity scores of 109 and 103 respectively on test data set.

When we train the PCFG model alone, the perplexity score on test data is 678. Combining the PCFG model with Good-Turing back-off and Kneser-Ney smoothing trigram models by linear interpolation, we obtain the test perplexity score 109 and 102 respectively. Next we train the PLSA model alone where the number of hidden semantic nodes h is set to be $|\mathcal{H}| = 125$, we obtain perplexity score on test data 1487. When this PLSA model is combined with Good-Turing back-off and Kneser-Ney smoothing trigram models by linear interpolation, we find that the test perplexity scores remain unchanged. If we combine these three models together using linear interpolation, we obtain the perplexity scores on test data 108 and 102 respectively.

Next we introduce the composite syntactic/trigram model which is equivalent to the composite syntactic/semantic/trigram language model by setting the semantic node h to be a constant. Using the generalized inside-outside algorithm to train this composite syntactic/trigram model with Good-Turing back-off and Kneser-Ney smoothing trigram models, we achieve a perplexity scores of 94 and 90 on test data of, a 14% and 11% relative reduction in perplexity respectively.

We then introduce the composite semantic/trigram model, which is equivalent to the composite syntactic/semantic/trigram language model by setting the syntatic node A to be a constant. We fix the number of possible hidden topics to be $|\mathcal{H}| = 125$ and use the generalized inside-outside algorithm to train the composite semantic/trigram model with Good-Turing back-off and Kneser-Ney smoothing trigram models. Here we achieve perplexity scores of 96 and 91 on test data, a 12% and 10% relative reduction in perplexity respectively. Since the representation for a document of the test data is not contained in the original training corpus, during testing we use "fold-in" heuristic approach similar to the one used in (Hofmann, 2001): the document-semantic parameters are re-estimated by maximum likelihood estimation while holding semantic-word parameters fixed, where the empirical distribution is given by the current updated document history.

Finally we use the generalized inside-outside algorithm to train the composite trigram/syntactic/semantic model with Good-Turing back-off and Kneser-Ney smoothing trigram models and we set the number of hidden semantic node h is again set to be $|\mathcal{H}| = 125$. Again since the representation for a document of the test data is not contained in the original training corpus, during testing we use "fold-in" heuristic approach as described in the last subsection: the document-semantic parameters are re-estimated by recursive gradient ascent of maximum likelihood estimation of the initial subsequence of a sentence while holding semantic-word and production rule parameters fixed. This time we achieve perplexity scores of 82 and 79 on test data, a 25% and 21% relative reduction in perplexity respectively.

Table 1. Perplexity results for the composite syntactic semantic trigram model on test corpus.

LANGUAGE MODEL	PERPLEXITY GOOD-TURING	PERPLEXITY KNESER-NEY
TRIGRAM (BASELINE)	109	103
PCFG	678	
PLSA	1487	
SYNTACTIC TRIGRAM	94	90
SEMANTIC TRIGRAM	96	91
SYNTACTIC, SEMANTIC TRIGRAM	82	79

The perplexity results are listed in Table 1 and the perplexity reductions of these results over baseline trigram models with Good-Turing and Kneser-Ney smoothings are shown in Figure 3. It shows that linear interpolation is too blunt to capture subtleties of PCFG and PLSA models, however our approach of integrating syntactic and semantic sources of nonlocal dependency information from PCFG and PLSA models into trigram model results significant perplexity improvement. Basically PCFG and PLSA models carry complementary long-range dependency structure and their gains over trigram model are almost additive. Another observation is that the gains of using Kneser-Ney smoothing over Good-Turing smoothing are almost additive too.

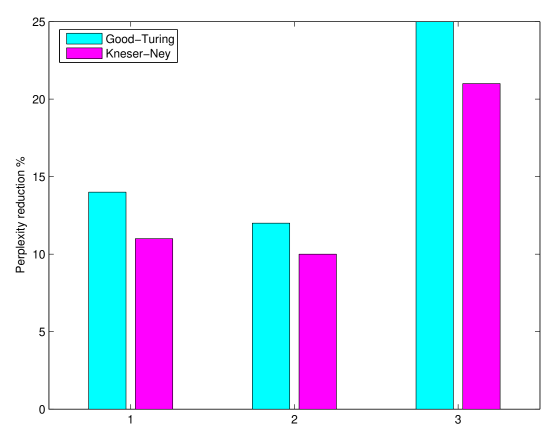


Figure 3. Relative perplexity reductions over baseline trigram with Good-Turing and Kneser-Ney smoothings by various composite language moddels: 1. Syntactic-trigram, 2. Semantic-trigram, 3. Syntactic-semantic-trigram.

6. Conclusion and Discussion

We present an original approach that combines n-gram, PCFG and PLSA to build a sophisticated mixed chain/tree/table directed MRF model for statistical language modeling, where various aspects of natural language—such as local word interaction, syntactic structure, and semantic document information—can be modeled by mixtures of exponential families with a rich expressive power that can take their interactions into account simultaneously and automatically. The composite directed MRF model we build becomes context sensitive grammar, and problems induced seem to be NP hard. However for this particular model, we show that we can generalize the well-known inside-outside algorithm to estimate its parameters in cubic time. To alleviate the sparseness of n-gram counts, we also generalize various smoothing techniques to handle cases where there exist hidden variables. The experiments we have carried out show improvement in perplexity over current state-of-the-art technique.

Griffiths et al. (2004) recently proposed a generative composite HMM/LDA (latent Dirichlet allocation) model which takes into account of both local sentence level syntactic class structures and global document level semantic contents for purposes of part-of-speech tagging and document classification, they have used MCMC to estimate the parameters for a much simpler model. However we propose an exact estimation algorithm for a much more complicated model.

One way to improve the quality of the language models is to use semantic smoothing (Bellegarda 2000, Wang et al. 2005), which has been shown to be effective in improving the perplexity results. Basically we can introduce an additional node between each topic node and word node to capture semantic similarity and subtle variation between words or introduce additional node between the topic nodes and the document node to take into account of semantic similarity and sub-topic variation within each document and among documents.

Blei et al. (2003) state that PLSA is not a well-defined generative model of documents, and there is no natural way to represent a document not seen in the original training corpus, this is why the "fold-in" heuristic procedure has to be used during testing to reestimate the semantic content. Blei et al. proposed LDA model to overcome this problem. It would be interesting to integrate LDA model into our composite language model and in this case variational method may have to be used.

References

Baker, J. (1979). Trainable grammars for speech recognition. *Proceedings of the 97th Meeting of the Acoustical Society of America*, 547-550.

Bellegarda, J. (2000). Exploiting latent semantic information in statistical language modeling. *Proceedings of IEEE*, 88(8):1279-1296.

Berger, A., Della Pietra, S. and Della Pietra, V. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39-71.

Blei, D., Ng, A. and Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993-1022.

Chelba, C. and Jelinek, F. (2000). Structured language modeling. *Computer Speech and Language*, 14(4):283-332.

Chen, S. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4): 319-358.

Chi, Z. (1999). Statistical properties of probabilistic context-free grammars. *Computational Linguistics*, 25(1):131-160.

Dempster, A., Laird, N. and Rubin, D. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1-38.

Griffiths, T., Steyvers, M., Blei, D. and Tenenbaum, J. (2004). Integrating topics and syntax. *Advances in Neural Information Processing Systems* 17.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177-196.

Jelinek, F. (1998). *Statistical Methods for Speech Recognition*. MIT Press.

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400-401.

Khudanpur, S. and Wu, J. (2000). Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. *Computer Speech and Language*, 14(4):355-372.

Lafferty, J. (2000). A derivation of the inside-outside algorithm from the EM algorithm. *IBM Research Report* 21636.

Pritchard, J., Stephens, M. and Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics*, 155:945-959.

Roark, B. (2001). Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249-276.

Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10(2):187-228.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(2):379-423.

Wainwright, M. and Jordan, M. (2003). Graphical models, exponential families, and variational inference. *Technical Report 649, Department of Statistics, UC-Berkeley*.

Wang, S., Schuurmans, D. and Zhao, Y. (2003). The latent maximum entropy principle. *Manuscript*.

Wang, S., Schuurmans, D., Peng, F. and Zhao, Y. (2005). Combining statistical language models via the latent maximum entropy principle. *Machine Learning*, 59:1-22.

Yedidia, S., Freeman, W. and Weiss, Y. (2001). Generalized belief propagation. *Advances in Neural Information Processing Systems*, 13:689-695.

Younger, D. (1967). Recognition and parsing of context free languages in time N^3. *Information and Control*, 10:198-208.