Treebank Transfer

Martin Jansche
Center for Computational Learning Systems
Columbia University
New York, NY 10027, USA
jansche@acm.org

Abstract

We introduce a method for transferring annotation from a syntactically annotated corpus in a source language to a target language. Our approach assumes only that an (unannotated) text corpus exists for the target language, and does not require that the parameters of the mapping between the two languages are known. We outline a general probabilistic approach based on Data Augmentation, discuss the algorithmic challenges, and present a novel algorithm for sampling from a posterior distribution over trees.

1 Introduction

Annotated corpora are valuable resources for Natural Language Processing (NLP) which often require significant effort to create. Syntactically annotated corpora – *treebanks*, for short – currently exist for a small number of languages; but for the vast majority of the world's languages, treebanks are unavailable and unlikely to be created any time soon.

The situation is especially difficult for dialectal variants of many languages. A prominent example is Arabic: syntactically annotated corpora exist for the common written variety (Modern Standard Arabic or MSA), but the spoken regional dialects have a lower status in written communication and lack annotated resources. This lack of dialect treebanks hampers the development of syntax-based NLP tools, such as parsers, for Arabic dialects.

On the bright side, there exist very large annotated (Maamouri et al., 2003, 2004a,b) corpora for Modern Standard Arabic. Furthermore, unannotated text corpora for the various Arabic dialects can also be assembled from various sources on the Internet. Finally, the syntactic differences between the Arabic dialects and Modern Standard Arabic are relatively minor (compared with the lexical, phonological, and morphological differences). The overall research question is then how to combine and exploit these resources and properties to facilitate, and perhaps even automate, the creation of syntactically annotated corpora for the Arabic dialects.

We describe a general approach to this problem, which we call *treebank transfer*: the goal is to project an existing treebank, which exists in a source language, to a target language which lacks annotated resources. The approach we describe is not tied in any way to Arabic, though for the sake of concreteness one may equate the source language with Modern Standard Arabic and the target language with a dialect such as Egyptian Colloquial Arabic.

We link the two kinds of resources that are available – a treebank for the source language and an unannotated text corpus for the target language – in a generative probability model. Specifically, we construct a joint distribution over source-language trees, target-language trees, as well as parameters, and draw inferences by iterative simulation. This allows us to impute target-language trees, which can then be used to train target-language parsers and other NLP components.

Our approach does not require aligned data, unlike related proposals for transferring annotations from one language to another. For example, Yarowksy and Ngai (2001) consider the transfer of word-level annotation (part-of-speech labels and bracketed NPs). Their approach is based on aligned

Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT), pages 74–82,
Vancouver, October 2005.

corpora and only transfers annotation, as opposed to generating the raw data plus annotation as in our approach.

We describe the underlying probability model of our approach in Section 2 and discuss issues pertaining to simulation and inference in Section 3. Sampling from the posterior distribution of target-language trees is one of the key problems in iterative simulation for this model. We present a novel sampling algorithm in Section 4. Finally in Section 5 we summarize our approach in its full generality.

2 The Probability Model

Our approach assumes that two kinds of resources are available: a source-language treebank, and a target-language text corpus. This is a realistic assumption, which is applicable to many source-language/target-language pairs. Furthermore, some knowledge of the mapping between source-language syntax and target-language syntax needs to be incorporated into the model. Parallel corpora are not required, but may help when constructing this mapping.

We view the source-language treebank as a sequence of trees $S_1,\ldots,S_n$, and assume that these trees are generated by a common process from a corresponding sequence of latent target-language trees $T_1,\ldots,T_n$. The parameter vector of the process which maps target-language trees to source-language trees will be denoted by Ξ. The mapping itself is expressed as a conditional probability distribution $p(S_i \mid T_i,\Xi)$ over source-language trees. The parameter vector Ξ is assumed to be generated according to a prior distribution $p(\Xi \mid \xi)$ with hyper-parameter ξ, assumed to be fixed and known.

We further assume that each target-language tree T_i is generated from a common language model Λ for the target language, $p(T_i \mid \Lambda)$. For expository reasons we assume that Λ is a bigram language model over the terminal yield (also known as the *fringe*) of T_i. Generalizations to higher-order n-gram models are completely straightforward; more general models that can be expressed as stochastic finite automata are also possible, as discussed in Section 5. Let $t_1,\ldots,t_k$ be the terminal yield of tree T. Then

$$p(T \mid \Lambda) = \Lambda(t_1 \mid \#) \left(\prod_{j=2}^{k} \Lambda(t_j \mid t_{j-1}) \right) \Lambda(\$ \mid t_k),$$

where # marks the beginning of the string and \$ marks the end of the string.

There are two options for incorporating the language model Λ into the overall probability model. In the first case – which we call the *full model* – Λ is generated by an informative prior distribution $p(\Lambda \mid \lambda)$ with hyper-parameter λ. In the second case – the *reduced model* – the language model Λ is fixed.

The structure of the full model is specified graphically in Figure 1. In a directed acyclic graphical model such as this one, we equate vertices with random variables. Directed edges are said to go from a parent to a child node. Each vertex depends directly on all of its parents. Any particular vertex is conditionally independent from all other vertices given its parents, children, and the parents of its children.

The portion of the full model we are interested in is the following factored distribution, as specified by Figure 1:

$$\begin{aligned} &p(S_1,\ldots,S_n,T_1,\ldots,T_n,\Lambda,\Xi \mid \lambda,\xi) \\ &= p(\Lambda \mid \lambda)\, p(\Xi \mid \xi) \prod_{i=1}^{n} p(T_i \mid \Lambda)\, p(S_i \mid T_i,\Xi) \end{aligned} \quad (1)$$

In the reduced model, we drop the leftmost term/vertex, corresponding to the prior for Λ with hyper-parameter λ, and condition on Λ instead:

$$\begin{aligned} &p(S_1,\ldots,S_n,T_1,\ldots,T_n,\Xi \mid \Lambda,\xi) \\ &\qquad = p(\Xi \mid \xi) \prod_{i=1}^{n} p(T_i \mid \Lambda)\, p(S_i \mid T_i,\Xi) \end{aligned} \quad (2)$$

The difference between the full model (1) and the reduced model (2) is that the reduced model assumes that the language model Λ is fixed and will not be informed by the latent target-language trees T_i. This is an entirely reasonable assumption in a situation where the target-language text corpus is much larger than the source-language treebank. This will typically be the case, since it is usually very easy to collect large corpora of unannotated text which exceed the largest existing annotated corpora by several orders of magnitude. When a sufficiently large target-language text corpus is available, Λ is simply a smoothed bigram model which is estimated once from the target-language corpus.

If the target-language corpus is relatively small, then the bigram model Λ can be refined on the basis of the imputed target-language trees. A bigram

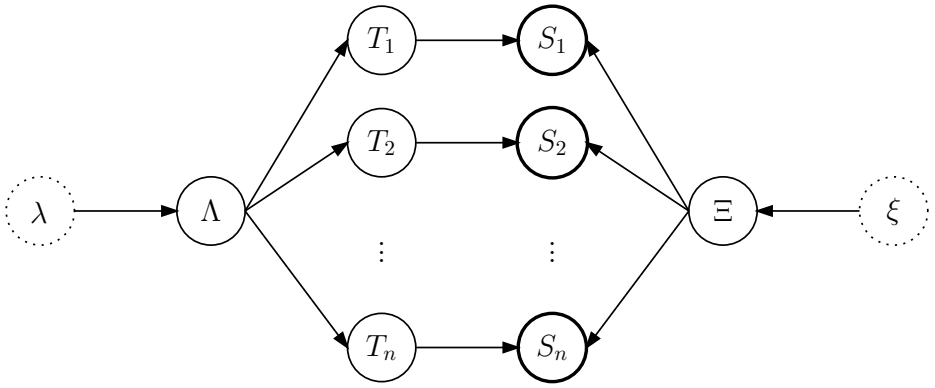

Figure 1: The graphical structure of the full probability model. Bold circles indicate observed variables, dotted circles indicate parameters.

model is simply a discrete collection of multinomial distributions. A simple prior for Λ takes the form of a product of Dirichlet distributions, so that the hyper-parameter λ is a vector of bigram counts. In the full model (1), we assume λ is fixed and set it to the observed bigram counts (plus a constant) in the target-language text corpus. This gives us an informative prior for Λ. If the bigram counts are sufficiently large, Λ will be fully determined by this informative prior distribution, and the reduced model (2) can be used instead.

By contrast, usually very little is known a priori about the syntactic transfer model Ξ. Instead Ξ needs to be estimated from data. We assume that Ξ too is a discrete collection of multinomial distributions, governed by Dirichlet priors. However, unlike in the case of Λ, the priors for Ξ are noninformative. This is not a problem, since a lot of information about the target language is provided by the language model Λ.

As one can see in Figure 1 and equation (1), the overall probability model constrains the latent target-language trees T_i in two ways: From the left, the language model Λ serves as a prior distribution over target-language trees. On the one hand, Λ is an informative prior, based on large bigram counts obtained from the target-language text corpus; on the other hand, it only informs us about the fringe of the target-language trees and has very little directly to say about their syntactic structure. From the right, the observed source-language trees constrain the latent target-language trees in a complementary fashion. Each target-language tree T_i gives rise to a corresponding source-language tree S_i according to the syntactic transfer mapping Ξ. This mapping is initially known only qualitatively, and comes with a noninformative prior distribution.

Our goal is now to simultaneously estimate the transfer parameter Ξ and impute the latent trees T_i. This is simplified by the following observation: if $T_1, \ldots, T_n$ are known, then finding Ξ is easy; vice versa, if Ξ is known, then finding T_i is easy. Simultaneous inference for Ξ and $T_1, \ldots, T_n$ is possible via Data Augmentation (Tanner and Wong, 1987), or, more generally, Gibbs sampling (Geman and Geman, 1984).

3 Simulation of the Joint Posterior Distribution

We now discuss the simulation of the joint posterior distribution over the latent trees $T_1, \ldots, T_n$, the transfer model parameter Ξ, and the language model parameter Λ. This joint posterior is derived from the overall full probability model (1). Using the reduced model (2) instead of the full model amounts to simply omitting Λ from the joint posterior. We will deal primarily with the more general full model in this section, since the simplification which results in the reduced model will be straightforward.

The posterior distribution we focus on is $p(T_1, \ldots, T_n, \Lambda, \Xi \mid S_1, \ldots, S_n, \lambda, \xi)$, which provides us with information about all the variables of interest, including the latent target-language trees T_i, the syntactic transfer model Ξ, and the target-language

language model Λ. It is possible to simulate this joint posterior distribution using simple sampling-based approaches (Gelfand and Smith, 1990), which are instances of the general Markov-chain Monte Carlo method (see, for example, Liu, 2001).

Posterior simulation proceeds iteratively, as follows. In each iteration we draw the three kinds of random variables – latent trees, language model parameters, and transfer model parameters – from their conditional distributions while holding the values of all other variables fixed. Specifically:

- Initialize Λ and Ξ by drawing each from its prior distribution.
- Iterate the following three steps:
  1. Draw each T_i from its posterior distribution given S_i, Λ, and Ξ.
  2. Draw Λ from its posterior distribution given $T_1, \ldots, T_n$ and λ.
  3. Draw Ξ from its posterior distribution given $S_1, \ldots, S_n$, $T_1, \ldots, T_n$, and ξ.

This simulation converges in the sense that the draws of $T_1, \ldots, T_n$, Λ, and Ξ converge in distribution to the joint posterior distribution over those variables. Further details can be found, for example, in Liu, 2001, as well as the references cited above.

We assume that the bigram model Λ is a family of multinomial distributions, and we write $\Lambda(t_j \mid t_{j-1})$ for the probability of the word t_j following t_{j-1}. Using creative notation, $\Lambda(\cdot \mid t_{j-1})$ can be seen as a multinomial distribution. Its conjugate prior is a Dirichlet distribution whose parameter vector λ_w are the counts of words types occurring immediately after the word type w of t_{j-1}. Under the conventional assumptions of exchangeability and independence, the prior distribution for Λ is just a product of Dirichlet priors. Since we employ a conjugate prior, the posterior distribution of Λ

$$p(\Lambda \mid S_1, \ldots, S_n, T_1, \ldots, T_n, \Xi, \lambda, \xi) = p(\Lambda \mid T_1, \ldots, T_n, \lambda) \quad (3)$$

has the same form as the prior – it is likewise a product of Dirichlet distributions. In fact, for each word type w the posterior Dirichlet density has parameter $\lambda_w + c_w$, where λ_w is the parameter of the prior distribution and c_w is a vector of counts for all word forms appearing immediately after w along the fringe of the imputed trees.

We make similar assumptions about the syntactic transfer model Ξ and its posterior distribution, which is

$$p(\Xi \mid S_1, \ldots, S_n, T_1, \ldots, T_n, \Lambda, \lambda, \xi) = p(\Xi \mid S_1, \ldots, S_n, T_1, \ldots, T_n, \xi). \quad (4)$$

In particular, we assume that syntactic transfer involves only multinomial distributions, so that the prior and posterior for Ξ are products of Dirichlet distributions. This means that sampling Λ and Ξ from their posterior distributions is straightforward.

The difficult part is the first step in each scan of the Gibbs sampler, which involves sampling each target-language latent tree from the corresponding posterior distribution. For a particular tree T_j, the posterior takes the following form:

$$\begin{aligned} p(T_j \mid S_1, \ldots, S_n, T_1, \ldots, T_{j-1}, T_{j+1}, \ldots, T_n, \Lambda, \Xi, \lambda, \xi) \\ = p(T_j \mid S_j, \Lambda, \Xi) = \frac{p(T_j, S_j \mid \Lambda, \Xi)}{\sum_{T_j} p(T_j, S_j \mid \Lambda, \Xi)} \\ \propto p(T_j \mid \Lambda)\, p(S_j \mid T_j, \Xi) \end{aligned} \quad (5)$$

The next section discusses sampling from this posterior distribution in the context of a concrete example and presents an algorithmic solution.

4 Sampling from the Latent Tree Posterior

We are faced with the problem of sampling T_j from its posterior distribution, which is proportional to the product of its language model prior $p(T_j \mid \Lambda)$ and transfer model likelihood $p(S_j \mid T_j, \Xi)$. Rejection sampling using the prior as the proposal distribution will not work, for two reasons: first, the prior is only defined on the yield of a tree and there are potentially very many tree structures with the same fringe; second, even if the first problem could be overcome, it is unlikely that a random draw from an n-gram prior would result in a target-language tree that corresponds to a particular source-language tree, as the prior has no knowledge of the source-language tree.

Fortunately, efficient direct sampling from the latent tree posterior is possible, under one very reasonable assumption: the set of all target-language trees which map to a given source-language tree S_j

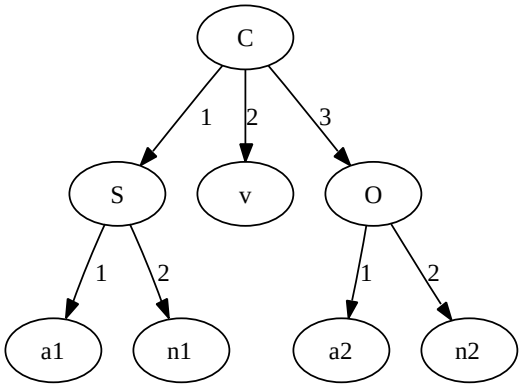

Figure 2: Syntax tree illustrating SVO constituent order within a sentence, and prenominal adjectives within noun phrases.

should be finite and representable as a packed forest. More specifically, we assume that there is a compact (polynomial space) representation of potentially exponentially many trees. Moreover, each tree in the packed forest has an associated weight, corresponding to its likelihood under the syntactic transfer model.

If we rescale the weights of the packed forest so that it becomes a normalized probabilistic context-free grammar (PCFG), we can sample from this new distribution (corresponding to the normalized likelihood) efficiently. For example, it is then possible to use the PCFG as a proposal distribution for rejection sampling.

However, we can go even further and sample from the latent tree posterior directly. The key idea is to intersect the packed forest with the *n*-gram language model and then to normalize the resulting augmented forest. The intersection operation is a special case of the intersection construction for context-free grammars and finite automata (Bar-Hillel et al., 1961, pp. 171–172). We illustrate it here for a bigram language model.

Consider the tree in Figure 2 and assume it is a source-language tree, whose root is a clause (C) which consists of a subject (S), verb (v) and object (O). The subject and object are noun phrases consisting of an adjective (a) and a noun (n). For simplicity, we treat the part-of-speech labels (a, n, v) as terminal symbols and add numbers to distinguish multiple occurrences. The syntactic transfer model is stated as a conditional probability distribution over source-language trees conditional on target language trees. Syntactic transfer amounts to independently changing the order of the subject, verb, and object, and changing the order of adjectives and nouns, for example as follows:

$$
\begin{aligned}
p(SvO \mid SvO) &= \Xi_1 \\
p(SOv \mid SvO) &= (1-\Xi_1)\,\Xi_2 \\
p(vSO \mid SvO) &= (1-\Xi_1)\,(1-\Xi_2)
\end{aligned}
$$

$$
\begin{aligned}
p(SvO \mid SOv) &= \Xi_3 \\
p(SOv \mid SOv) &= (1-\Xi_3)\,\Xi_4 \\
p(vSO \mid SOv) &= (1-\Xi_3)\,(1-\Xi_4)
\end{aligned}
$$

$$
\begin{aligned}
p(SvO \mid vSO) &= \Xi_5 \\
p(SOv \mid vSO) &= (1-\Xi_5)\,\Xi_6 \\
p(vSO \mid vSO) &= (1-\Xi_5)\,(1-\Xi_6)
\end{aligned}
$$

$$
\begin{aligned}
p(an \mid an) &= \Xi_7 \\
p(na \mid an) &= 1-\Xi_7
\end{aligned}
$$

$$
\begin{aligned}
p(an \mid na) &= \Xi_8 \\
p(na \mid na) &= 1-\Xi_8
\end{aligned}
$$

Under this transfer model, the likelihood of a target-language tree $[_A\, v\, [_S\, a_1\, n_1]\, [_O\, n_2\, a_2]]$ corresponding to the source-language tree shown in Figure 2 is $\Xi_5 \times \Xi_7 \times \Xi_8$. It is easy to construct a packed forest of all target-language trees with non-zero likelihood that give rise to the source-language tree in Figure 2. Such a forest is shown in Figure 3. Forest nodes are shown as ellipses, choice points as rectangles connected by dashed lines. A forest node is to be understood as an (unordered) disjunction of the choice points directly underneath it, and a choice point as an (ordered, as indicated by numbers) conjunction of the forest nodes directly underneath it. In other words, a packed forest can be viewed as an acyclic and-or graph, where choice points represent and-nodes (whose children are ordered). As a simplifying convention, for nodes that dominate a single choice node, that choice node is not shown. The forest in Figure 3 represents *SvO*, *SOv*, and *vSO* permutations at the sentence level and *an*, *na* permutations below the two noun phrases. The twelve overall permutations are represented compactly in terms of two choices for the subject, two choices for the object, and three choices for the root clause.

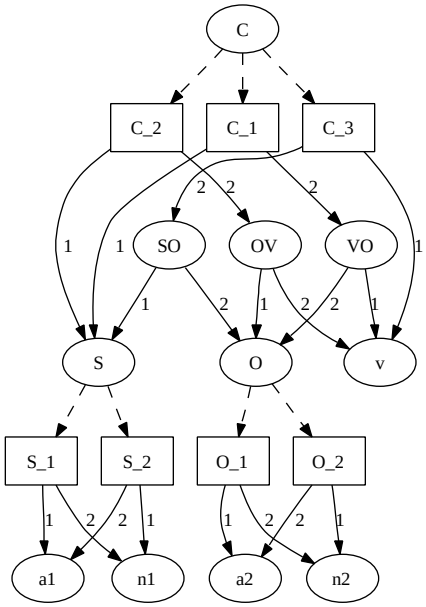

Figure 3: Plain forest of target-language trees that can correspond to the source-language tree in Figure 2.

We intersect/compose the packed forest with the bigram language model Λ by augmenting each node in the forest with a left context word and a right peripheral word: a node N is transformed into a triple (a,N,b) that dominates those trees which N dominates in the original forest and which can occur after a word a and end with a word b. The algorithm is roughly[1] as shown in Figure 5 for binary branching forests; it requires memoization (not shown) to be efficient. The generalization to forests with arbitrary branching factors is straightforward, but the presentation of that algorithm less so. At the root level, we call `forest_composition` with a left context of # (indicating the start of the string) and add dummy nodes of the form $(a,\$,\$)$ (indicating the end of the string). Further details can be found in the prototype implementation. Each node in the original forest is augmented with two words; if there are n leaf nodes in the original forest, the total number of nodes in the augmented forest will be at most n^2 times larger than in the original forest. This means that the compact encoding property of the packed forest (exponentially many trees can be represented in polynomial space) is preserved by the composition algorithm. An example of composing a packed forest with a bigram language model appears in Figure 4, which shows the forest that results from composing the forest in Figure 3 with a bigram language model.

The result of the composition is an augmented forest from which sampling is almost trivial. The first thing we have to do is to recursively propagate weights from the leaves upwards to the root of the forest and associate them with nodes. In the non-recursive case of leaf nodes, their weights are provided by the bigram score of the augmented forest: observe that leaves in the augmented forest have labels of the form (a,b,b), where a and b are terminal symbols, and a represents the immediately preceding left context. The score of such a leaf is simply $\Lambda(b \mid a)$. There are two recursive cases: For choice nodes (and-nodes), their weight is the product of the weights of the node's children times a local likelihood score. For example, the node (v,O,n) in Figure 4 dominates a single choice node (not shown, per the earlier conventions), whose weight is $\Lambda(a \mid v)\,\Lambda(n \mid a)\,\Xi_7$. For other forest nodes (or-nodes), their weight is the sum of the weights of the node's children (choice nodes).

Given this very natural weight-propagation algorithm (and-nodes correspond to multiplication, or-nodes to summation), it is clear that the weight of the root node is the sum total of the weights of all trees in the forest, where the weight of a tree is the prod-

[1] A detailed implementation is available from `http://www.cs.columbia.edu/~jansche/transfer/`.

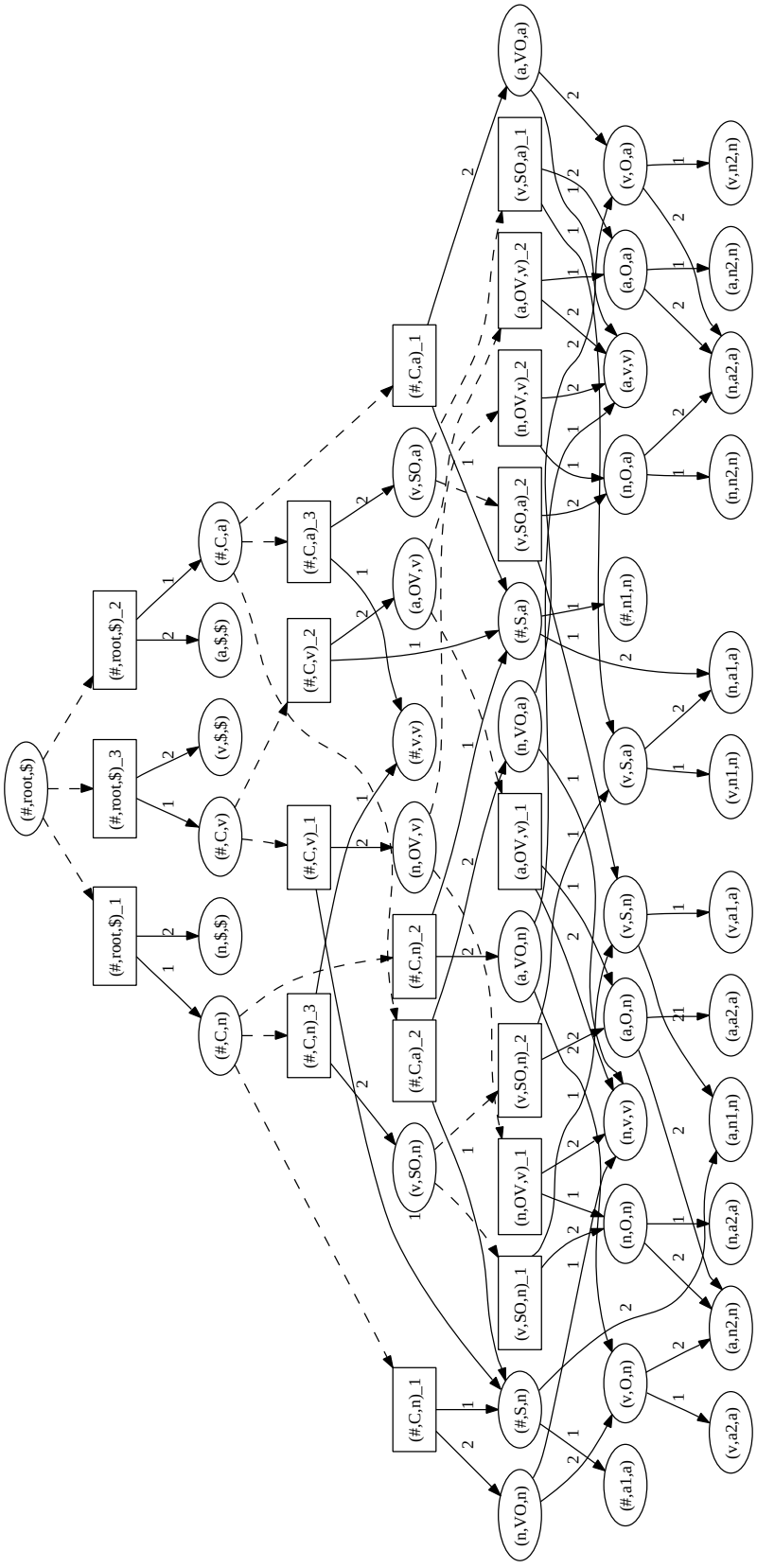

Figure 4: Augmented forest obtained by intersecting the forest in Figure 3 with a bigram language model.

```
forest_composition(N, a):
  if N is a terminal:
    return { (a,N,N) }
  else:
    nodes = {}
    for each (L,R) in N.choices:
      left_nodes <- forest_composition(L, a)
      for each (a,L,b) in left_nodes:
        right_nodes <- forest_composition(R, b)
        for each (b,R,c) in right_nodes:
          new_n = (a,N,c)
          nodes <- nodes + { new_n }
          new_n.choices <- new_n.choices + [((a,L,b), (b,R,c))]
    return nodes
```

Figure 5: Algorithm for computing the intersection of a binary forest with a bigram language model.

uct of the local likelihood scores times the language model score of the tree's terminal yield. We can then associate outgoing normalized weights with the children (choice points) of each or-node, where the probability of going to a particular choice node from a given or-node is equal to the weight of the choice node divided by the weight of the or-node.

This means we have managed to calculate the normalizing constant of the latent tree posterior (5) without enumerating the individual trees in the forest. Normalization ensures that we can sample from the augmented and normalized forest efficiently, by proceeding recursively in a top-down fashion, picking a child of an or-node at random with probability proportional to the outgoing weight of that choice. It is easy to see (by a telescoping product argument) that by multiplying together the probabilities of each such choice we obtain the posterior probability of a latent tree. We thus have a method for sampling latent trees efficiently from their posterior distribution.

The sampling procedure described here is very similar to the lattice-based generation procedure with *n*-gram rescoring developed by Langkilde (2000), and is in fact based on the same intersection construction (Langkilde seems to be unaware that the CFG-intersection construction from (Bar-Hillel et al., 1961) is involved). However, Langkilde is interested in optimization (finding the best tree in the forest), which allows her to prune away less probable trees from the composed forest in a procedure that combines composition, rescoring, and pruning. Alternatively, for a somewhat different but related formulation of the probability model, the sampling method developed by Mark et al. (1992) can be used. However, its efficiency is not well understood.

5 Conclusions

The approach described in this paper was illustrated using very simple examples. The simplicity of the exposition should not obscure the full generality of our approach: it is applicable in the following situations:

- A prior over latent trees is defined in terms of stochastic finite automata.

  We have described the special case of bigram models, and pointed out how our approach will generalize to higher-order *n*-gram models. However, priors are not generally constrained to be *n*-gram models; in fact, any stochastic finite automaton can be employed as a prior, since the intersection of context-free grammars and finite automata is well-defined. However, the intersection construction that appears to be necessary for sampling from the posterior distribution over latent trees may be rather cumbersome when higher-order *n*-gram models or more complex finite automata are used as priors.

- The inverse image of an observed tree under the mapping from latent trees to observed trees can be expressed in terms of a finite context-free language, or equivalently, a packed forest.

The purpose of Gibbs sampling is to simulate the posterior distribution of the unobserved variables in the model. As the sampling procedure converges, knowledge contained in the informative but structurally weak prior Λ is effectively passed to the syntactic transfer model Ξ. Once the sampling procedure has converged to a stationary distribution, we can run it for as many additional iterations as we want and sample the imputed target-language trees. Those trees can then be collected in a treebank, thus creating novel syntactically annotated data in the target language, which can be used for further processing in syntax-based NLP tasks.

Acknowledgements

I would like to thank Steven Abney, the participants of the 2005 Johns Hopkins workshop on Arabic dialect parsing, and the anonymous reviewers for helpful discussions. The usual disclaimers apply.

References

Y. Bar-Hillel, M. Perles, and E. Shamir. 1961. On formal properties of simple phrase structure grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14(2):143–172.

Alan E. Gelfand and Adrian F. M. Smith. 1990. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Matching and Machine Intelligence*, 6(6):721–741.

Irene Langkilde. 2000. Forest-based statistical sentence generation. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 170–177. ACL Anthology A00-2023.

Jun S. Liu. 2001. *Monte Carlo Strategies in Scientific Computing*. Springer.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Hubert Jin. 2004a. Arabic Treebank: Part 2 v 2.0. Electronic resource, available from LDC.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Hubert Jin. 2004b. Arabic Treebank: Part 3 v 1.0. Electronic resource, available from LDC.

Mohamed Maamouri, Ann Bies, Hubert Jin, and Tim Buckwalter. 2003. Arabic Treebank: Part 1 v 2.0. Electronic resource, available from LDC.

Kevin Mark, Michael Miller, Ulf Grenander, and Steve Abney. 1992. Parameter estimation for constrained context-free language models. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23–26, 1992*, pages 146–149. ACL Anthology H92-1028.

Martin A. Tanner and Wing Hung Wong. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.

David Yarowksy and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 200–207.