UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

A Bayesian Model for Recursive Recovery of Syntactic Dependencies

Permalink

https://escholarship.org/uc/item/5v172907

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 27(27)

ISSN

1069-7977

Authors

Phaf, R. Hans
Rotteveel, Mark
Wendte, Robert

Publication Date

2005

Peer reviewed

A Bayesian Model for Recursive Recovery of Syntactic Dependencies

Virginia Savova (savova@jhu.edu)
Department of Cognitive Science, 3300 N. Charles St
Johns Hopkins University, Baltimore, MD 21218 USA

Leonid Peshkin (pesha@hms.harvard.edu)
Department of Systems Biology, 200 Longwood Ave
Harvard University, Boston, MA 02115 USA

Abstract

Bayesian inference on graphical models can account for a variety of psychlogical data in non-linguistic domains. Recent proposals have touted its biological plausibility, which raises the question to what extent it may capture the learning and use of grammar. We propose a way to structure the parsing task in order to make it amenable to local classification methods. This allows us to build a Dynamic Bayesian Network which uncovers the syntactic dependency structure of English sentences. Experiments with a small test corpus demonstrate that the model successfully learns from labeled data. We discuss what this approach may tell us about the way syntax may be encoded in the brain and about the modularity of the language faculty.

Keywords

Parsing, Finite-state models, Bayesian networks, Biological plausibility, Dependency Grammars, Grammar induction, Cognitive modeling.

Introduction

Bayesian graphical models have become an important explanatory strategy in cognitive science ([Knill and Richards, 1996], [Kording and Wolpert, 2004], [Stocker and Simoncelli, 2005]). Recent work strongly supports their biological plausibility in general and that of dynamic Bayesian models in particular [Rao, 2005]. Dynamic models are geared towards prediction and classification of sequences. As such, they are naturally suitable for language modeling and have already been aplied to tasks like speech recognition [Livescu et al., 2003] and part-of-speech tagging [Peshkin et al., 2003]. However, grammar learning and parsing with such models generally appears out-of-reach, because of their Markovian character.

Markov models restrict possible dependencies to a bounded, local context. At one extreme, the context is confined to the symbol occupying the current position in the sequence (order-0 or unigram models). In more relaxed versions, context may include a fixed number of positions before the current symbol (k-order), typically no more than two (trigram models). The restricted space of possible dependencies allows transition probabilities to be infered from the data and stored in a look-up table with relatively little technical sophistication.

Not surprisingly however, the restricted space of representable dependencies is also the main disadvantage of Markov models in syntax-related tasks like parsing. Syntactic dependencies in natural language are unboundedly non-local, in the sense that no fixed amount of context is guaranteed to contain the members of a given constituent. For example, consider the sentences in examples (1 - 3). In the first sentence, the subject *king* and verb *bought* are adjacent to one another. Thus, the dependecy between them would be captured by a bigram (order-1) model. However, the same model would be unable to represent the dependendency in the second example, because the subject and verb are separated by two words. To capture this dependency, we need a 2nd-order Markov model (trigram). Similarly, the 2nd-order model would prove inadequate for the third example, where the subject and verb are separated by four words.

(1) The king bought a camel.

(2) The king of Prussia bought a camel.

(3) The king of some strange country bought a camel.

Our solution to this problem relies on representing sentences with non-local dependencies like (2, 3) as derived from their local dependency variants, akin to (1). This intuition is based on the formal notion that a string with non-local dependency is obtained from a dependency tree via a recursive linearization procedure. The string obtained at each step of the linearization procedure contains new local dependencies, which push apart local dependencies from previous levels. This way of conceptualizing the linearization of syntactic structure allows us to use a Dynamic Bayesian Network despite its Markov properties. We construct a DBN parser which decides only on local attachments. We then call the parser recursively to uncover the underlying dependency tree. Our results show that the model captures grammatical knowledge for all levels of the derivation. The biological plausibility and remarkable compactness of learned representation may suggest that parsing in the brain is accomplished in a similar manner.

Dependency grammar

Tree-based linguistic representations of natural language syntax treat non-local dependencies as local in the two-dimensional tree structure, of which the string is a one dimensional projection. The dependency grammar representation of (1) captures the dependency between the

subject, the object and the verb, and the dependency between the determiners and their respective nouns (Figure 1).
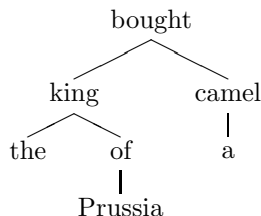
bought
king   camel
the   of   a
Prussia

Figure 1: Dependency structure of example (2)

More formally, a dependency grammar consists of a lexicon of terminal symbols (words), and an inventory of dependency relations specifying inter-lexical requirements. A string is generated by a dependency grammar if and only if:

- Every word but one (ROOT) is dependent on another word.
- No word is dependent on itself either directly or indirectly.
- No word is dependent on more than one word.
- Dependencies do not cross.

In a dependency tree, each word is the mother of its dependents, otherwise known as their HEAD. To linearize the dependency tree in Figure 1 into a string, we introduce the dependents recursively next to their heads:
Step I: bought
Step II: king bought camel
Step III: The king of bought a camel
Step IV: The king of Prussia bought a camel.

Recursive parsing as local classification

Parsing in the dependency grammar framework is the task of uncovering the dependency tree given the sentence. Suppose that instead of searching for a complete parse given a complete sentence, we restricted our task to compressing the string up the linearization path. Note that linearization is essentially dependency parsing in reverse. In other words, we can uncover the dependency structure by labeling the **local** head-dependent relationships at the bottom linearization level (i.e. the sentence) and erasing from the string the words whose heads are already found. We recursively process the output until the root level. Thus, if as a first step in parsing (2), we pick the head of *Prussia* to be the preposition *of*, we can compress the string to a form virtually equivalent to linearization Step III. Picking *king* as the head of the preposition leads us to compress the string further, to the equivalent of step II. To compress the string, we must simply identify which words in the string occupy a position adjacent to their heads.

The attractive feature of this representation is that the parsing decisions taken at each step are local. Hence, parsing can be converted into a local classification task. The task is to chose the best sequence of labels denoting

local dependency relationships (links). At each position, we choose between setting the link to LEFT, RIGHT, or NONE, where LEFT/RIGHT means the word is dependent on its left/right neighbor. NONE means the search for this word's head should be postponed until later stages of compression. The output of the classifier is a labeled string, which can be compressed by removing linked dependents. It is fed through recursively, until the string is compressed to the ROOT.

The Dynamic Bayesian Network classifier

The first step towards building the classifier is coming up with a feature representation. We will briefly motivate the choice of feature set with linguistic arguments. It is easy to determine that the linking pattern of a word depends on its part of speech (PoS) and the part of speech of its neighbor. For example, English determiners only link to the right, and adverbs link almost exclusively to verbs. However, the parts of speech alone are not sufficient to determine linking behavior. In some cases, the identity of the adjacent word is required - *bought* accepts links from nouns to the right, while *slept* does not.

Another decisive factor is how many dependents the current word has acquired so far. Since once the current word is linked it will become unavailable as a future linking target to other words, it is important to acertain that its valency has already been satisfied. Valency refers to the **minimal** number of dependents a word actively seeks to license. In English and other SVO languages, the word has particular requirements with respect to the number of left and right dependents. Thus, in our feature representation, valency is indirectly captured by two variables, which reflect the number of dependents which had already been linked to the current word from either side - LEFT and RIGHT COMPOSITE (COMP). The COMP variables affect not only the linking behavior of the current token, but that of its neighbor as well. If the word has already received many dependents from one side, the probability of accepting yet another one becomes smaller, since its valency is already satisfied.

Finally, the current label depends on the labels of its neighbor, because if the previous label is RIGHT, then the current label cannot be LEFT, and if the next label is LEFT, the current label cannot be RIGHT. Thus, our full feature representation consists of the word and its PoS tag, the words and PoS tags of its neighbor, the two valencies of the current word, the right valency of its left neighbor and the left valency of its right neighbor, as well as the neighboring links.

The Word and Next Word feature vocabulary contain the 2500 most frequent words in the data. An additional value was allocated for all remaining out-of-vocabulary words. The PoS, and Next PoS vocabulary contain 36 of the original 45 Penn Treebank Tagset, after all punctuation PoS tags were removed. The left and right COMP features had tree values: NONE, ONE and MANY.

This feature representation is used as the basis of the Dynamic Bayesian Network (DBN). After we briefly introduce the essential aspects of DBNs, we wil expand on the structure of the network for parsing. For more infor-

mation on the general class of models, we refer the reader to a recent dissertation [Murphy, 2002] for an excellent survey.

General notes on DBNs

A DBN is a Bayesian network unwrapped in "time" (i.e. over a sequence), such that it can represent dependencies between variables at adjacent position. More formally, a DBN consists of two models B^0 and B^+, where B^0 defines the initial distribution over the variables at position 0, by specifying:

- set of variables $X_1, \ldots, X_n$;

- directed acyclic graph over the variables;

- for each variable X_i a table specifying the conditional probability of X_i given its parents in the graph $Pr(X_i|Par\{X_i\})$.

The joint probability distribution over the initial state is:

$$\Pr(X_1, ..., X_n) = \prod_1^n \Pr(X_i|Par\{X_i\}).$$

The transition model B^+ specifies the conditional probability distribution (CPD) over the state at time t given the state at time $t-1$. B^+ consists of:

- directed acyclic graph over the variables $X_1, \ldots, X_n$ and their predecessors $X_1^-, \ldots, X_n^-$ — roots of this graph;

- conditional probability tables $\Pr(X_i|Par\{X_i\})$ for all X_i (but not X_i^-).

The transition probability distribution is:

$$\Pr(X_1, ..., X_n \Big| X_1^-, ..., X_n^-) = \prod_1^n \Pr(X_i|Par\{X_i\}).$$

Together, B^0 and B^+ define a probability distribution over the realizations of a system through time, which justifies calling these BNs "dynamic". In our setting, the word's index in a sentence corresponds to time, while realizations of a system correspond to correctly tagged English sentences. Probabilistic reasoning about such system constitutes inference.

Standard inference algorithms for DBNs are similar to those for HMMs. Note that, while the kind of DBN we consider could be converted into an equivalent HMM, that would render the inference intractable due to a huge resulting state space. In a DBN, some of the variables will typically be observed, while others will be hidden. The typical inference task is to determine the probability distribution over the states of a hidden variable over time, given time series data of the observed variables. This is usually accomplished using the forward-backward algorithm. Alternatively, we might obtain the most likely sequence of hidden variables using the Viterbi algorithm. These two kinds of inference yield resulting LINK tags.
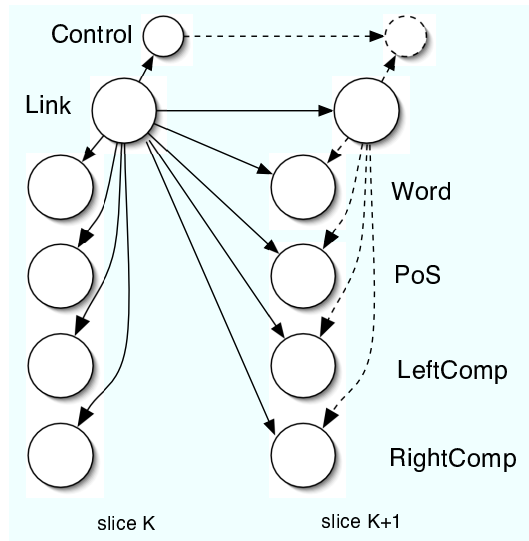

Figure 2: The parsing DBN.

Learning the parameters of a DBN from data is generally accomplished using the EM algorithm. However, in our model, learning is equivalent to collecting statistics over cooccurrences of feature values and link labels. This is implemented in GAWK scripts and takes minutes on a large corpus. While in large DBNs, exact inference algorithms are intractable, and are replaced by a variety of approximate methods, the number of hidden state variables in our model is small enough to allow exact algorithms to work. For the inference we use the standard algorithms, as implemented in a recently released toolkit [Bilmes and Zweig, 2002].

Structure of the DBN parser

Each slice of our DBN parser is a representation of the joint probability distribution of WORD, POS, LEFT/RIGHT COMP, and the hidden variable LINK Figure 2. In our model, the link determines the value of all variables and they are independent of one another. Of course, this is not truly the case, but among those variables LINK is the only unobserved, hence modeling all other dependencies is inconsequential. In addition to the intra-slice dependencies, we model dependencies between the current, previous and next position. The LINK variable infulences all aforementioned variables in neighboring slices. Finally, we introduce a CONTROL variable which deterministically ensures that at least one link in the sequence will be set to something other than NONE. This forces the parser to trully compress the string at each recursive parsing step.

Experiments and results

For the results presented here we used the WSJ10 corpus [Klein and Manning, 2004]. It is a subset of the WSJ Penn Treebank ([Marcus et al., 1993]), consisting of all sentences shorter than eleven words with punctuation

removed [1]. Eliminating the punctuation was done to simulate parsing in the oral modality. The dependency annotation was obtained through automatic conversion of the original treebank annotation. The relatively short sentences make this corpus a good approximation to casual speech and limit the effects of misattachments due to the conversion. Note that the parser is in principle capable of handling longer sentences.

Encoding

The corpus was encoded in our feature representation as follows. For each sentence, a number of feature files were produced containing the feature representation of the sentence at each linearization level. The encoding of an actual sentence-structure pair from our corpus (Figure 3), is illustrated in Figures 4 to 7.
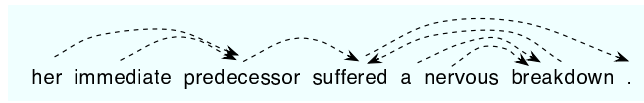

Figure 3: Dependency structure.

At the lowest level, no word has any discovered dependents, hence the COMP values are zero everywhere. All links of words whose heads are not adjacent are labeled NONE (0).

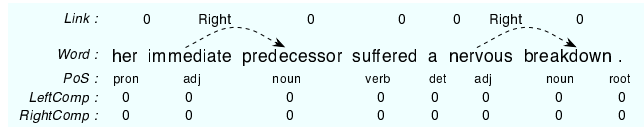At the next level, words whose labels were LEFT or

Figure 4: First layer representation.

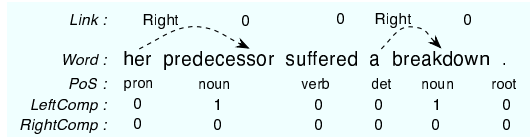RIGHT are removed from the structure and the COMP counters for their head are incremented.

Figure 5: Second layer representation.

The same procedure produces the subsequent levels (Figures 6, 7)

Testing

The corpus was split randomly 9:1 into a training and testing section. In training mode, the DBN was given

---

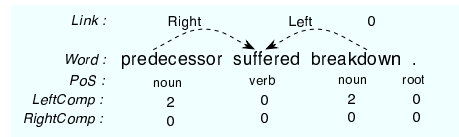[1]the dot in our figures stands for an abstract ROOT symbol
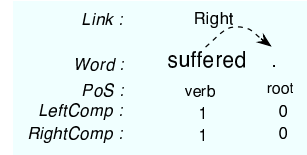

Figure 6: Third layer representation.

Figure 7: Top layer representation.

all levels with the correct labels. It was trained directly on the annotations, with no additional smoothing. The result achieved was 79% correct link attachment for directed dependencies, and 82% for undirected. We compare the results to two baselines given for this corpus by [Klein and Manning, 2004], Table 1.

Table 1: DBN results against baseline.

Model	Accuracy	
	Dir	Undir
DBN	79	82
Random	30	46
Adjacent heuristic	34	57

More detailed results for our model are shown in Table 2 . The results unequivocally surpass the random baseline, and the best available heuristic, which amounts to linking every word to its right neighbor. This suggests our model has learned at least some of the non-trivial dependencies which govern the choice of link structure. The minimal difference between the vocabulary and out-of-vocabulary scores imply that the network can recover the syntactic properties of an unknown word in context. The fact that the root accuracy is higher than the non-root accuracy allows us to conclude that the network correctly learns to postpone decisions about the root word in all cases, and about its dependent in most cases.

Table 2: Detailed results for the DBN.

Measure	Accuracy
Root dependency	83
Non-root dependency	78
Out-of-Vocabulary	75
Sentence	36

Discussion

Our results show that combining a DBN model with recursive application is a reasonable parsing strategy. This opens the door to the hypothesis that Bayesian inference is a possible mechanism for parsing in the brain, despite the Markovian properties of the corresponding dynamic models. The high ROOT accuracy suggests that the model has captured some fundamental principles defining the local dependency structure at all levels of the derivation. We take this result as evidence that graphical models with Markov properties are capable of handling unbounded non-local dependencies through recursive calls on their own output. The implication of this finding transcend Bayesian graphical models and speak to the general issue of how relevant other biologically plausible Markov models can be to language processing and learning. For example, Elman networks have been criticized for their a priori limitation in handling unbounded dependencies [Frank et al., 2005]. It is possible that such type of models may be adapted to discover locality in the hierarchical structure through recursive application.

One exciting implication of this hypothesis is the domain-generality of Bayesian inference and learning mechanisms. Previous work has proposed that these mechanisms are involved in visual perception [Knill and Richards, 1996], [Kersten and Yuille, 2003], motor control [Kording and Wolpert, 2004], and attention modulation [Yu and Dayan, 2005]. [Kersten and Yuille, 2003] proposes Bayesian graphical model of object detection which rely on estimating hidden variables such as relative depth and 3-D structure from observables they influence -shadow displacement, 2-D projection. [Kording and Wolpert, 2004] suggests that subjects ina sensory-motor experiment internally represent both the statistical distribution of the task and their sensory uncertainty, combining them in a manner consistent with a performance-optimizing bayesian process. In our work, the hidden links are estimated from observable word and PoS, along with a prior label distibution.

The parallelism in the proposed cognitive strategies for all these different modalities may shed light on the issue whether and how modular the language faculty is. The modularity hypothesis states that the cognitive mechanisms underlying linguistic competence are specific to language. If Bayesian inference proves to be a plausible uniting principle behind visual, motor and linguistic abilities, this hypothesis is seriously undermined. At the same time, it is important to note that the generality of the mechanism does not necessarily negate the modularity of language completely. The feature representation which our model used already encodes language-specific knowledge. Further research is needed to determine whether the feature representation and the structure of the network can be induced through structure learning algorithms.

Our approach is particularly appealing in light of recent work suggesting that Bayesian type inference is biologically plausible. [Rao, 2005] shows that recurrent networks of noisy integrate-and-fire neurons can perform approximate Bayesian inference for dynamic and hierarchical graphical models. According to him, the membrane potential dynamics of neurons corresponds to approximate belief propagation in the log domain, and the spiking probability of a neuron approximates the posterior probability of the preferred state encoded by the neuron, given past inputs. This seems to suggest that our parsing model can be implemented in a neural circuit. Furthermore, since the same DBN is used to uncover local dependencies throughout all levels of the derivation, such implementation would address Humboldt's characterization of language as a system that makes "infinite use of finite means" at the neurophysiological level. The same neural aparatus could be used to recursively uncover the dependency structure of a sentence level by level.

Another implication of our work is that the nature of the processing architecture may constrain the kind of grammar human languages permit. If indeeed parsing is accomplished through recursive processing of the output of previous stages, some types of long-distance depndencies would be impossible to detect. In particular, if the material intervening between a head-dependent pair (H, D) is not a constituent whose own head depend on either H or D, our model would not be able to uncover it because H and D will not be adjacent at any point in the derivation. In other words, this parser is incapable of handling strictly context-sensitive languages. to the extent that such dependencies exist, they are fairly limited [Shieber, 1985]. Such cases will need to be resolved through some reordering in pre-processing, possibly based on case marking.

Future work

One deficiency of our model is that decisions at lower levels cannot be reversed in the interest of more optimal choices at higher levels. There are however important reasons why this might be necessary. For example, a prepositional phrase subcategorized for by the verb may be mistakenly attached to a preceding noun phrase, leaving the verb with a missing dependent (4)

(4)   The king put *[the camel in the trunk].

In the future, we hope to address this problem through a form of beam search - retaining the k-best parses at each level and choosing among them based on what happens at the next level.

Another important issue that we need to address is the total loss of information about the dependents that have been linked to a word at previous levels. Some well-known cases pose a problem for this aspect of our model. For example, the sentences in (6) and (5) are structurally distinct solely becase the complement of the prepositional phrase in the second sentence is an instrument appropriate for seeing.

(5)   The king saw [the camel with two humps] .

(6)   The king saw *[the camel with a telescope].

In our current model, once the complement is linked to the preposition, the two sentence will become identical, and one of them will be assigned the wrong structure. This concern can be addressed through introducing new variables, which keep track not only of the number of linked dependents but of their semantic category (e.g. instrument, animate etc.)

A natural way to extend our model in a different direction is to combine it with the Bayesian PoS tagger developed in [Peshkin et al., 2003]. Allowing the model to infer PoS tags and structure simultaneously will be a significantly better approximation to the parsing task humans are faced with. Last but not least, we would like to implement semisupervised learning. One way to do this would involve starting off with a small labeled set of sentences at all parsing depths, followed by presenting unparsed whole sentences. The parses suggested by the model would in their turn be used for learning in a bootstrap fashion.

Conclusion

In our closing remarks, we would like to emphasize several aspects of our parsing model which make it interesting from the perspective of cognitive science. First, it belongs to a class of models which have been used recently to capture cognitive mechanisms in non-linguistic domains. Second, it naturally utilizes the overwhelming "disguised locality" of natural language syntax - in other words, it benefits from the fact that string-non-local dependencies are tree-local. Third, it is biologically plausible because it has been shown to be implementable in a neural circuit. And finally, it takes seriously the question how the finite amount of brain hardware is capable of encoding structures of unbounded depth. While there is much room for improvement, we believe these qualities make it an important step on the difficult road toward understanding how the mind emerges from the brain.

References

[Bilmes and Zweig, 2002] Bilmes, J. and Zweig, G. (2002). The graphical models toolkit: An open source software system for speech and time-series processing. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, June 2002. Orlando Florida.

[Frank et al., 2005] Frank, R., Mathis, D., and Badecker, W. (2005). The acquisition of anaphora by simple recurrent networks. unpublished manuscript.

[Kersten and Yuille, 2003] Kersten, D. and Yuille, A. (2003). Bayesian models of object perception. *Cognitive Neuroscience*, 13:1–9.

[Klein and Manning, 2004] Klein, D. and Manning, C. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the ACL*.

[Knill and Richards, 1996] Knill, D. C. and Richards, W., editors (1996). *Perception as Bayesian inference*. Cambridge University Press.

[Kording and Wolpert, 2004] Kording, K. and Wolpert, D. (2004). Bayesian integration in sensorimotor learning. *Nature*, 427:244–247.

[Livescu et al., 2003] Livescu, K., Glass, J., and Bilmes, J. (2003). Hidden feature models for speech recognition using dynamic bayesian networks. In *8th European Conference on Speech Communication and Technology (Eurospeech)*.

[Marcus et al., 1993] Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19.

[Murphy, 2002] Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, Univ. of California at Berkeley.

[Peshkin et al., 2003] Peshkin, L., Pfeffer, A., and Savova, V. (2003). Bayesian nets for syntactic categorization of novel words. In *Proceedings of the NAACL*.

[Rao, 2005] Rao, R. P. N. (2005). Hierarchical bayesian inference in networks of spiking neurons. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA.

[Shieber, 1985] Shieber, S. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.

[Stocker and Simoncelli, 2005] Stocker, A. and Simoncelli, E. (2005). Constraining a bayesian model of human visual speed perception. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA.

[Yu and Dayan, 2005] Yu, A. J. and Dayan, P. (2005). Inference, attention, and decision in a bayesian neural architecture. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA.