Turning the pipeline into a loop: Iterated unsupervised dependency parsing and PoS induction

Christos Christodoulopoulos[†], **Sharon Goldwater**[‡], **Mark Steedman**[‡]
School of Informatics
University of Edinburgh
[†]christos.c@ed.ac.uk [‡]{steedman,sgwater}@inf.ed.ac.uk

1 Motivation

Most unsupervised dependency systems rely on gold-standard Part-of-Speech (PoS) tags, either directly, using the PoS tags instead of words, or indirectly in the back-off mechanism of fully lexicalized models (Headden et al., 2009).

It has been shown in supervised systems that using a hierarchical syntactic structure model can produce competitive sequence models; in other words that a parser can be a good tagger (Li et al., 2011; Auli and Lopez, 2011; Cohen et al., 2011). This is unsurprising, as the parser uses a rich set of hierarchical features that enable it to look at a less localized environment than a PoS tagger which in most cases relies solely on local contextual features. However this interaction has not been shown for the unsupervised setting. To our knowledge, this work is the first to show that using dependencies for unsupervised PoS induction is indeed useful.

2 Iterated learning

Although most unsupervised systems depend on gold-standard PoS information, they can also be used in a fully unsupervised pipeline. One reason for doing so is to use dependency parsing as an extrinsic evaluation for unsupervised PoS induction (Headden et al., 2008). As discussed in that paper (and also by Klein and Manning (2004)) the quality of the dependencies drops with the use of induced tags. One way of producing better PoS tags is to use the dependency parser's output to influence the PoS inducer, thus turning the pipeline into a loop.

The main difficulty of this approach is to find a way of incorporating dependency information into a PoS induction system. In previous work (Christodoulopoulos et al., 2011) we have described BMMM: a PoS induction system that makes it is easy to incorporate multiple features either at the type or token level. For the dependency induction system we chose the DMV model of Klein and Manning (2004) because of its simplicity and its popularity. Both systems are described briefly in section 3.

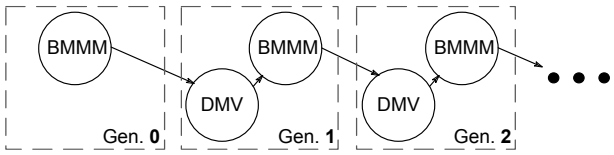


Figure 1: The iterated learning paradigm for inducing both PoS tags and dependencies.

Using these two systems we performed an *iterated learning* experiment. The term is borrowed from the language evolution literature meaning "the process by which the output of one individual's learning becomes the input to other individuals' learning" (Smith et al., 2003). Here we treat the two systems as the individuals[1] that influence each other in successive generations starting from a run of the original BMMM system without dependency information (fig. 1). We start with a run of the basic BMMM system using just context and morphology features (generation 0) and use the output to train the DMV. To complete the first generation, we then use the induced dependencies as features (as described in section 4) for a new run of the BMMM system.

As there is no single objective function, this setup

[1]This is not directly analogous to the language evolution notion of iterated learning; here instead of a single type of individual we have two separate systems that learn/model different representations.

NAACL-HLT Workshop on the Induction of Linguistic Structure, pages 96–99,
Montréal, Canada, June 3-8, 2012.

does not guarantee that either the quality of PoS tags or the dependencies will improve after each generation. However, in practice this iterated learning approach works well (as we discuss in section 4).

3 Component models

3.1 DMV model

The basic DMV model (Klein and Manning, 2004) generates dependency trees based on three decisions (represented by three probability distributions) for a given head node: whether to attach children in the left or right direction; whether or not to stop attaching more children in the specific direction given the adjacency of the child in that direction; and finally whether to attach a specific child node. The probability of an entire sentence is the sum of the probabilities of all the possible derivations headed by ROOT.

The DMV model can be seen as (and is equivalent to) a Context Free Grammar (CFG) with only a few rules from head nodes to generated children and therefore the model parameters can be estimated using the Inside-Outside algorithm (Baker, 1979).

3.2 BMMM model

The Bayesian Multinomial Mixture Model (Christodoulopoulos et al., 2011), illustrated in figure 2, assumes that all tokens of a given word type belong to a single syntactic class, and each type is associated with a number of features (e.g., morphological or contextual features), which form the observed variables. The generative process first chooses a hidden class z for each word type and then chooses values for each of the observed features of that word type, conditioned on the class. Both the distribution over classes θ and the distributions over each kind of feature $\phi^{(t)}$ are multinomials drawn from Dirichlet priors α and $\beta^{(t)}$ respectively. A main advantage of this model is its ability to easily incorporate features either at the type or token level; as in Christodoulopoulos et al. (2011) we assume a single type-level feature m (morphology, drawn from $\phi^{(m)}$) and several token-level features $f^{(1)} \dots f^{(T)}$ (e.g., left and right context words and, in our extension, dependency features).

Inference in the model is performed using a collapsed Gibbs sampler, integrating out the model parameters and sampling the class label z_j for each word type j from the following posterior distribution:

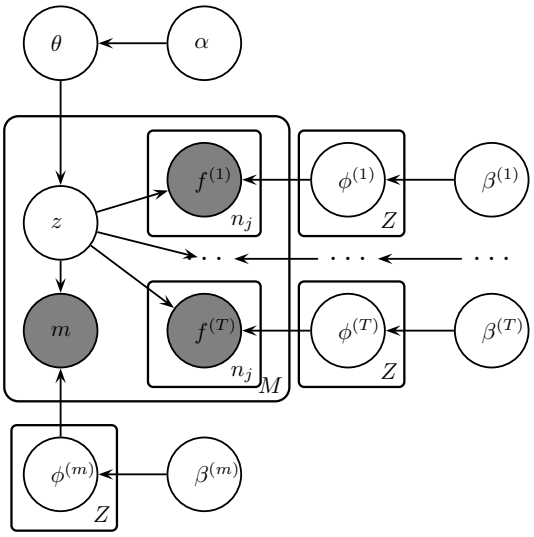


Figure 2: The BMMM with T kinds of token-level features ($f^{(t)}$ variables) and a single kind of type-level feature (morphology, m). M is the total number of word types, Z the number of classes, and n_j the number of tokens of type j.

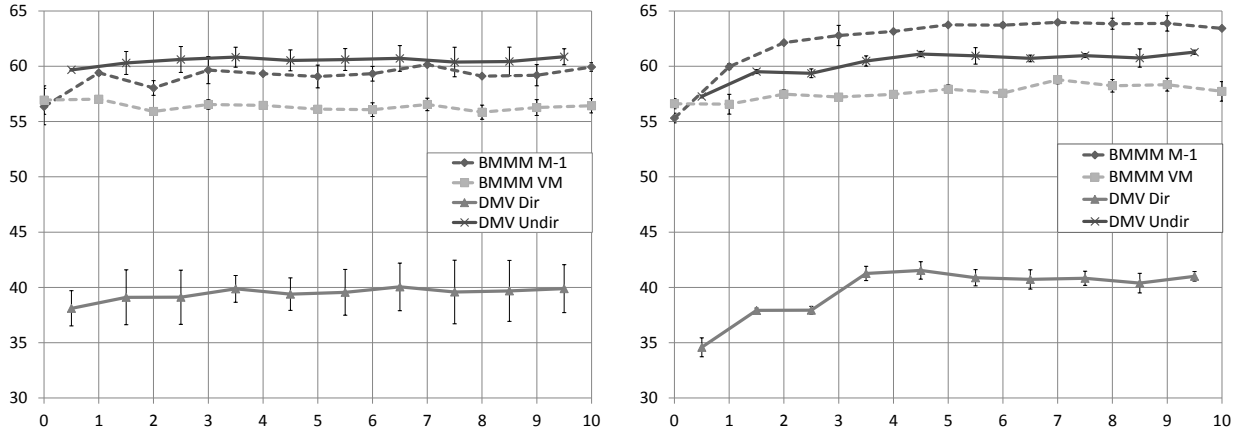
$$P(z_j \mid \mathbf{z}_{-j}, \mathbf{f}, \alpha, \beta) \propto P(z_j \mid \mathbf{z}_{-j}, \alpha, \beta) P(\mathbf{f}_j \mid \mathbf{f}_{-j}, \mathbf{z}, \alpha, \beta) \quad (1)$$

where the first factor $P(z_j)$ is the prior distribution over classes (the mixing weights) and the second (likelihood) factor $P(\mathbf{f}_j)$ is the probability given class z_j of all the features associated with word type j. Since the different kinds of features are assumed to be independent, the likelihood can be rewritten as:

$$P(\mathbf{f}_j \mid \mathbf{f}_{-j}, \mathbf{z}, \alpha, \beta) = P(f_j^{(m)} \mid \mathbf{f}_{-j}^{(m)}, \mathbf{z}, \alpha, \beta) \cdot \prod_{t=1}^{T} P(\mathbf{f}_j^{(t)} \mid \mathbf{f}_{-j}^{(t)}, \mathbf{z}, \beta) \quad (2)$$

and, as explained in Christodoulopoulos et al. (2011), the joint probability of all the token level features of kind t for word type j is computed as:

$$P(\mathbf{f}_j^{(t)} \mid \mathbf{f}_{-j}^{(t)}, z_j = z, \mathbf{z}_{-j}, \beta) = \frac{\prod_{k=1}^{K^{(t)}} \prod_{i=0}^{n_{jk}-1} (n_{jk,z} + i + \beta)}{\prod_{i=0}^{n_j-1} (n_{\cdot,z} + i + F\beta)} \quad (3)$$



(a) Using only directed dependencies as features

(b) Using directed and undirected dependencies as features

Figure 3: Developmental results on WSJ10. The performance of the PoS inducer is shown in terms of many-to-1 accuracy (BMMM M1) and V-Measure accuracy (BMMM VM) and the performance of the dependency inducer is shown using directed and undirected dependency accuracy (DMV Dir and DMV Undir respectively).

where $K^{(t)}$ is the dimensionality of $\phi^{(t)}$ and n_{jk} is the number of instances of feature k in word type j.

4 Experimental design

Because the different kinds of features are assumed to be independent in the BMMM, it is easy to add more features into the model; this simply increases the number of factors in equation 2. To incorporate dependency information, we added a feature for word-word dependencies. In the model, this means that for a word type j with n_j tokens, we observe n_j dependency features (each being the head of one token of j). Like all other features, these are assumed to be drawn from a class-specific multinomial $\phi_z^{(d)}$ with a Dirichlet prior $\beta^{(d)}$.

Using lexicalized head dependencies introduces sparsity issues in much the same way contextual information does. In the case of context words, the BMMM and most vector-based clustering systems use a fixed number of most frequent words as features; however in the case of dependencies we use the induced PoS tags of the previous generation as grouping labels: we aggregate the head dependency counts of words that have the same PoS tag, so the dimension of $\phi_z^{(d)}$ is just the number of PoS tags.

The dependency features are used in tandem with the features used in the original BMMM system, namely the 100 most frequent context words (± 1 context window), the suffixes extracted from the Morfessor system (Creutz and Lagus, 2005) and the extended morphology features of Haghighi and Klein (2006).

For designing the iterated learning experiments, we used the 10-word version of the WSJ corpus (WSJ10) as development data and ran the iterative learning process for 10 generations. To evaluate the quality of the induced PoS tags we used the many-to-1 (M1) and V-Measure (VM) metrics and for the induced dependencies we used directed and undirected accuracy.

Figure 3a presents the developmental result of the iterated learning experiments on WSJ10 where only directed dependencies where used as features. We can see that although there was some improvement in the PoS induction score after the first generation, the rest of the metrics show no significant improvement throughout the experiment.

When we used undirected dependencies as features (figure 3b) the improvement over iterations was substantial: nearly 8.5% increase in M1 and 1.3% in VM after only 5 iterations. We can also see that the results of the DMV parser are improving as well: 7% increase in directed and 3.8% in undirected accuracy. This is to be expected, since as Headden et al. (2008) show, there is a (weak) correlation between the intrinsic scores of a PoS inducer and the

performance of an unsupervised dependency parser trained on the inducer's output.

Using the same development set we selected the remaining system parameters; for the BMMM we fixed the number of induced classes to the number of gold-standard PoS tags for each language and used 500 sampling iterations with annealing. For the DMV model we used 20 EM iterations. Finally we used observed that after 5 generations the rate of improvement seems to level, so for the rest of the languages we use only 5 learning iterations.

References

Michael Auli and Adam Lopez. 2011. A comparison of loopy belief propagation and dual decomposition for integrated CCG supertagging and parsing. In *Proceedings of ACL-HLT*, pages 470–480.

James K. Baker. 1979. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132, June.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2011. A Bayesian mixture model for pos induction using multiple features. In *Proceedings of EMNLP*, pages 638–647, Edinburgh, Scotland, UK., July.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of EMNLP*, pages 50–61.

Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *In Proceedings of AKRR*, volume 5, pages 106–113.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of NAACL*, pages 320–327.

William P. Headden, David McClosky, and Eugene Charniak. 2008. Evaluating unsupervised part-of-speech tagging for grammar induction. In *Proceedings of COLING*, pages 329–336.

William P. Headden, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of NAACL*, pages 101–109.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of ACL*.

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for chinese POS tagging and dependency parsing. In *Proceedings of EMNLP*, pages 1180–1191.

Kenny Smith, Simon Kirby, and Henry Brighton. 2003. Iterated learning: a framework for the emergence of language. *Artif. Life*, 9(4):371–386.