Two Approaches for Building an Unsupervised Dependency Parser and their Other Applications

Jagadeesh Gorla, Amit Goyal and **Rajeev Sangal**
Language Technologies Research Centre
International Institute of Information Technology, Hyderabad, India
{jagadeesh@research.,amitgoyal@students., sangal@}iiit.net

Abstract

Much work has been done on building a parser for natural languages, but most of this work has concentrated on supervised parsing. Unsupervised parsing is a less explored area, and unsupervised *dependency* parser has hardly been tried. In this paper we present two approaches for building an unsupervised dependency parser. One approach is based on learning dependency relations and the other on learning subtrees. We also propose some other applications of these approaches.

Introduction

A parser for natural languages gives an analysis of the structure of a sentence. The analysis could be in terms of phrase structure or dependency structure. The parsers can also be trained. Many parsers are available which use human annotated data for learning. Collin's (Collins, M. J. 1999) parser is one of the best known examples of this. The problem is that for many languages very little annotated data is available. This makes unsupervised parsing an important area of research in Natural Language Processing (NLP). Some effort has been made in this direction (Klein, D. 2005; Bod 2006), but the unsupervised parsers tried so far are meant for analyzing phrase structure. Unsupervised dependency parsing has been neglected, partly because it is a harder problem. For applications like machine translation and information extraction, dependency parsers could be more useful than phrase structure parsers based on Context Free Grammar (CFG). As part of ongoing research, we are trying to build an unsupervised dependency parser using two different approaches, namely learning dependency relations and learning dependency subtrees. Each of these tries to learn the syntax of the language in a particular way. Since this is a very hard problem, it is possible that the unsupervised dependency parser built at the end may not be directly usable for practical applications. However, the approaches being proposed can be directly used for solving other problems.

One assumption we have made is that a part of speech (POS) tagger is available for the given language. We are not relying on any other language resource or tool except a sufficiently large unannotated corpus.

Preprocessing

For training the parser, a POS tagger is run on the unannotated corpus. The output, which consists of sequences of tokens, i.e., words and their POS tags (grammatical categories) is preprocessed such that only the tags are retained. One important exception is the preposition tag which is lexicalized. This, in a way, means that every preposition identified by the tagger is actually treated as a POS tag itself. Thus, the result of the preprocessing step consists of sequences POS tags and lexicalized prepositions. We do this because prepositions are the single most important closed class of words for determining dependency relations.

Measure of Association

Both the approaches used by us rely on a measure of association between distant word (or POS tag) pairs, i.e., two words occurring at a particular distance. The measure that we have used is a modified version of the mutual information measure. The modification takes into account the extra variable of distance. This measure can be called conditional mutual information (CMI), where the condition is the value of the distance. The value of CMI is calculated as:

$$CMI \;=\; p(x,\, y,\, d)\; log\; \frac{p(x,\, y,\, d)}{p(x)\; p(y)} \qquad (1)$$

where x, y are either POS tags or lexicalized prepositions and d is the distance between x and y.

CMI is then normalized to get an initial measure of dependency between two words at various distances. This normalized CMI is calculated as:

$$NCMI \;=\; \frac{CMI \;-\; min(CMI)}{max(CMI) \;-\; min(CMI)} \qquad (2)$$

Depending upon whether the absolute distance is considered or not, the $NCMI$ measure can be an estimate of undirected or directed dependency relation. If signed distance is considered than the assumption is that the order of the words indicates which word is the head. We have done some experiments on this measure of association and the results indicate that it is a good estimate of dependency.

Overview of Training

Using the sequences obtained after preprocessing, we calculate the mutual information between the tags and words with respect to distance. This is the initial measure of undirected or directed dependency relation, depending on whether the sign of the distance is considered or not and whether we make an assumption about word order determining the directions of relations or not.

In the next step, we construct a completely connected weighted graph for a given sentence using $NCMI$ as the weight. Then we find the top k maximum spanning trees. These spanning trees are sorted and ranked based on the total weight of the edges. For each spanning tree, since these are undirected trees, we take each node of these trees as the possible root and get the possible directed spanning trees, i.e., possible dependency trees. The rank of the dependency tree is kept the same as that of the spanning tree. These spanning trees are the candidate dependency trees.

Finally, we learn either the dependency relations or subtrees (Bod 2006), as described below.

Learning Dependency Relations

In this approach, we first weight each dependency relation in the dependency tree based on the rank of the tree. Then we learn the most probable dependency relation between tags based on the contextual parameters. The output of the training process will be dependency relations between the tags at a particular distance and the corresponding weight.

Learning Subtrees

Rens Bod (Bod 2006) has used the subtree learning approach for phrase structure. We will be using a variation of this approach for dependency trees. This will involve calculating all the possible subtrees of the spanning tree at all the levels of the tree. We then assign weights to the subtrees based on the rank of the dependency tree (directed spanning tree). The step of finding the top k maximum spanning trees and sorting and ranking them is repeated for all the sentences and a forest of subtrees is constructed. A probability is assigned to each subtree based on the rank, weight and the distribution of subtree in the forest. Unlike in Bod's method, we select spanning trees based on an initial measure of dependency, instead of selecting trees randomly.

Parsing

Once the system has been trained, we can use an adapted version of an existing technique for parsing sentences. For parsing based on dependency relations, we first construct a graph G with a dummy root, sentence words as the vertices with directed weighted edges. The weights of the edges are assigned based on the weights learnt during training. Finding the dependency tree is finding the maximum spanning tree (MST) of G (McDonald *et al.* 2005). To find non-projective dependency tree (MST), we are using Chi-Liu-Edmonds algorithm (Chu, Y. J. and Liu, T. H. 1965; Edmonds, J. 1967) and for projective dependency tree, we use Eisner's parsing algorithm (Eisner, J. 1996). We are also planning to use the k-best hypergraph parsing algorithm (Huang & Chiang 2005) for this purpose.

We will also try to apply Unsupervised Maximum Likelihood Data Oriented Parsing (UML-DOP) proposed by Bod (Bod 2006) for directly parsing sentences using the learnt subtrees.

Other Applications

Apart from using the two approaches mentioned above for unsupervised parsing, we will be working on other applications of the intermediate results obtained.

Augmenting Supervised Parsers

One obvious application is to use these results to augment supervised parsers to overcome the problem of sparsity of annotated data.

Error Correction for Machine Translated Text

Machine translated text has many errors because machine translation is a very hard problem and is still in its initial stages, at least for general purpose translation. The results obtained by us for dependency relations or learnt subtrees can be used to correct some of these errors.

Multi Word Expressions

We also plan to explore how learning subtrees or dependency relations can be used for identifying multi word expressions and to calculate their compositionality. Venkatapathy and Joshi (Venkatapathy & Joshi 2005) combined various features of dependency relations to rank them according to their compositionality. CMI and the dependency relations obtained from our work can be plugged into their system for measuring compositionality more accurately.

References

Bod, R. 2006. An all-subtrees approach to unsupervised parsing. In *Proceedings of COLING-ACL*.

Chu, Y. J. and Liu, T. H. 1965. On the shortest arborescence of a directed graph. In *Science Sininca*.

Collins, M. J. 1999. Head-driven statistical models for natural language parsing. In *PhD thesis, University of Pennsylvania*.

Edmonds, J. 1967. Optimum branchings. In *Journal of Research of the National Bureau of Standards*.

Eisner, J. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proc. of COLING*.

Huang, L., and Chiang, D. 2005. Better k-best parsing. In *Proceedings of IWPT*.

Klein, D. 2005. The unsupervised learning of natural language structure. In *Ph.D thesis, Stanford University*.

McDonald, R.; Pereira, F.; Ribarov, K.; and Hajic, J. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT-EMNLP*.

Venkatapathy, S., and Joshi, A. 2005. Measuring the relative compositionality of verb-noun (v-n) collocations by integrating features. In *Proceedings of HLT-EMNLP*.