This paper presents a framework for parsing dependency trees using spanning tree algorithms. Particularly, it works for both non-projective and projective dependency parsing. The problem is formalized as the search for a maximum spanning tree in a directed graph. Chu-Liu-Edmonds algorithm and Eisner algorithm are applied to find maximum spanning trees in directed graphs. Moreover, McDonald and coworkers’ online large-margin discriminative training method is used to learn weight vectors for the directed graphs.

In the discussion about dependency parsing and spanning trees, edge-based factorization is a building block, which is scored by the dot product between a high dimensional feature representation and a weight vector (learned by MIRA). And the total score for a dependency tree is the sum of the scores of all edges. Given that a dependency tree is a spanning tree, the task of finding one with highest score is equivalent to searching for a maximum spanning tree (MST) in a corresponding directed graph. For non-projective trees, there are no restrictions about crossing edges in the tree. Chu-Liu-Edmonds algorithm gives an efficient solution (O(n2) with good implementation). The algorithm has each vertex in the graph greedily select the incoming edge with highest weight. If a tree is generated, then it is guaranteed to be an MST. If not, there is a cycle, which will be contracted into a node with recalculated edge weights for relative edges. Thanks to the property that an MST on the contracted graph is equivalent to an MST in the original graph, the algorithm can call itself *recursively* on the new graph. In contrast, for projective trees, Eisner algorithm with bottom-up dynamic programming gives satisfactory performances. It maintains the nested structural constraint for projective trees, which is not necessary for non-projective trees. As a brief comparison, non-projective parsing is indeed easier than projective paring since non-crossing constraint is not enforced. Moreover, non-projective parsing has better time complexity (O(n2)) than projective parsing (O(n3)).

To learning weight vector in the directed graph for a sentence, margin infused relaxed algorithm (MIRA) serves as the basic idea. And the authors apply the work of McDonald and coworkers as extension of MIRA. During the update of the weight vector in training process, the algorithm “keeps the new value as close as possible to the old value while subjecting to correctly classifying the instance under consideration with a margin given by the loss of the incorrect classifications”, where the loss is the number of words being labeled with incorrect parents in dependency parsing task. Then there is a problem about exponential blow-up constraints. The authors discuss 2 solutions: One is single-best MIRA. It relaxes the optimization by using inly the single margin constraint with the highest score. And the other is factored MIRA. It claims that the weight of correct incoming edge for a word and the weight of all other incoming edges must be separated by a margin of “1”, which reduces the number of constraints to O(n2). However, since it gives generally more restrictive than the original constraints, the optimal solution might be ruled out in feasible sets.

Finally, some experiments are conducted to compare different algorithms. The dataset is the Czech Prague Dependency Treebank, the tasks contains part-of-speech tagging, and the metrics includes accuracy and complete (number of completely correct trees). The proposed algorithms give improvements on accuracy, which are more obvious on a subset of dataset with many non-projective dependencies. Moreover, since the model “sets its weights with respect to the parsing algorithm and will disfavor features over unlikely non-projective edges”, there is no problem of searching the entire space of non-projective trees for primarily projective languages (finding extremely bad trees).