Enhancing First-Pass Attachment Prediction

Fabrizio Costa [1], Paolo Frasconi [1] , Vincenzo Lombardo [2], Patrick Sturt [3], Giovanni Soda [1]

Abstract. This paper explores the convergence between cognitive modeling and engineering solutions to the parsing problem in NLP. Natural language presents many sources of ambiguity, and several theories of human parsing claim that ambiguity is resolved by using past (linguistic) experience. In this paper we analyze and refine a connectionist paradigm (Recursive Neural Networks) capable of processing acyclic graphs to perform supervised learning on syntactic trees extracted from a large corpus of parsed sentences. Following a widely accepted hypothesis in psycholinguistics, we assume an incremental parsing process (one word at a time) that keeps a connected partial parse tree at all times. By implementing a parsing simulation procedure, we collect a large amount of data that shows the viability of the RNN as informant of a disambiguation process. We analyze what kind of information is exploited by the connectionist system in order to resolve different sources of ambiguity, and we see how the generalization performance of the system is affected by the tree complexity and the frequency of specific subtrees. We finally propose some enhancements to the architecture in order to achieve a better prediction accuracy.

1 INTRODUCTION

Incremental processing of natural language (incrementality for short) is an intuitively plausible hypothesis upon the human language processor. The incrementality hypothesis has received a large experimental support in the psycholinguistic community over the years [5]. More recently, the hypothesis has been exploited in a few computational approaches to parsing. Lane & Henderson [8] propose Simple Synchrony Networks, an architecture that can learn to generate structural relationships between syntactic constituents and is employed to build a parse tree for a given input sentence. Roark & Johnson [10] devised an incremental parser based on production rules learned in the framework of stochastic context-free grammars. In its general form, the incrementality hypothesis implies that the semantic interpretation of some fragment of the sentence is available as the scan of the input material proceeds from left to right so that the syntactic analyzer has to keep a totally connected structure at all times (that is, the semantic interpreter has no means to work on disconnected items). Under this assumption, parsing proceeds from left to right through a sequence of trees, each spanning one more word to the right until completion of the sentence. The above notion of incrementality can be briefly formalized as follows. Given a sentence $s = w_0 \cdots w_{|s|-1}$ and a parse tree T, the incremental tree T_i spanning $w_0, \cdots, w_i$ is recursively defined as:

- the chain from w_0 to its maximal syntactic projection if $i = 0$, or
- T_{i-1} and the chain from w_i to N, where N is either a node of T_{i-1}, or the lowest node of T dominating both the root of T_{i-1} and w_i.

The *connection path* for w_i is the difference between the incremental trees T_i and T_{i-1}. In this framework, a (non lexicalized) incremental grammar is defined as the triple $(\mathcal{N}, \mathcal{P}, \mathcal{C})$ being $\mathcal{N}$ a set of nonterminals, $\mathcal{P}$ a set of POS (Part of Speech) tag symbols, and $\mathcal{C}$ a set of connection paths. In each connection path we distinguish two special nodes, called the *anchor* and the *foot*, annotated with symbols in $\mathcal{N}$ and $\mathcal{P}$, respectively. A derivation in this grammar is a sequence of attachment operations having the form $(T_{i-1}, cp_j) \rightarrow T_i$, where $cp_j \in \mathcal{C}$. We define the *right frontier* as the sequence obtained by taking the rightmost child of every node starting from the root and ending on the foot node. For an attachment to be admissible the following two conditions must hold: (1) if $i > 0$ there must exist a node v in the right frontier of T_{i-1} matching the anchor of cp_j, and (2) the foot of cp_j must be the POS tag of w_i. Tree T_i is then constructed by joining T_{i-1} and cp_j at v (see Fig.2). Although interesting because of its strong connection to human parsing, such a grammar is highly ambiguous. For example, after extracting $\mathcal{C}$ from a set of 40,000 parsed sentences in the Penn Treebank [9], the expected number of admissible attachments at a single word position is on average 126, although only one attachment is correct [4]. Costa et al. [3, 2] recently proposed the prediction of correct attachments by using a machine learning algorithm based on recursive neural networks [6]. They show that after training on a small corpus of 500 parsed sentences, the accuracy of prediction significantly outperforms common linguistic heuristics such as late closure and minimal attachment [7]. Moreover, the model effectively reproduces well known cognitive phenomena such as recency attachment for adverbs, relative clauses in English, closure ambiguities, and preference for NP over S in complement ambiguities [11]. The model proposed in [2] is briefly sketched in the next section. In the rest of this paper we extensively study the behavior of the model using a realistic corpus of about 42,000 parsed sentences from the Penn TreeBank. The aim of this investigation is twofold. Firstly we want to acquire a deeper understanding on the preferences expressed by the connectionist model (Section 4). Secondly we want to exploit this knowledge to enhance the network performance. In Section 5 we show how the results of the investigation enabled us to conceive two novel prediction algorithms, obtaining significant improvements in terms of accuracy and computational effort.

2 LEARNING FIRST-PASS ATTACHMENTS

Our problem is to specify a learning architecture which is capable of learning parsing decisions on the basis of incremental trees. Our

[1] Dept. of Systems and Computer Science, University of Florence, Firenze, Italy
[2] Dept. of Computer Science, Università di Torino, Italy
[3] HCRC, University of Glasgow, Glasgow, UK

[4] These attachments are referred to as ”first-pass” attachments in psycholinguistics, because there can be a ”second pass” due to structural revision.

approach relies on *Recursive Neural Networks* [6], a machine learning architecture which is capable of learning to classify hierarchical data structures, such as the incremental trees which we employ in this paper. The task of the model is to take any given word w_i and incremental tree T_{i-1}, and to rank the candidate incremental trees that can be produced by attaching w_i to T_{i-1}. The highest ranked tree will be chosen as the preferred alternative. More formally, each example is a pair (F_i,j^*), where F_i is the forest of alternatives (a bag of trees corresponding to all the incremental trees resulting from all the possible attachments of the current word w_i), and j^* is the index of the correct tree (T_i) in F_i. A recursive neural network is asked to predict the conditional probability that a tree T_i is the correct one, given the forest F_i:

$$y_{ij} = P(j = j^* | F_i) \text{ for each } j \in [1, |F|]$$

where j is the index of a tree in the forest, and y_{ij} is the probability estimated for the tree indexed j in the forest F_i. Assuming the correct tree belongs in the forest F_i (i.e., assuming the grammar induced from the database of all available connection paths is complete), the probabilities for all of the candidate trees must sum to 1 ($\sum_j y_{ij} = 1$). A hidden state vector $X(v) \in R^n$ is associated with each node v and encodes the subtree dominated by v. The dimension n must be large enough to give sufficient expressive power to the network. The state vector is computed by a state transition function which combines the state vectors of v's daughters with a vector encoding of the label of v. This function is computed by a multilayer perceptron, which is replicated at each node in the tree. In our case, however, we are evaluating a forest of alternative trees, not a single tree, requiring a slight variation over the standard recursive network described in [6]. In particular, we employ a recursive neural network to process each tree in the forest, and each network uses the same transition function and the same output function, with shared weights. Moreover, the output function is linear, yielding a real output a_{ij} associated with the j-th tree in F_i. All the linear outputs are finally transformed using the softmax function (normalized exponentials), yielding the estimates of the conditional probabilities y_{ij} for each tree in the candidate forest. The trees are then ranked according to this probability, to obtain the order of preference. Training maximizes the conditional log-likelihood of the predicted preferences, given the true tree j^*. Optimization is based on a gradient descent procedure, where gradients are computed by the back-propagation through structure algorithm [6].

3 DATA PREPARATION

Our results are based on the Wall Street Journal Section of the Penn-Treebank Corpus [9]. We have adopted the standard setting widely adopted in literature (see, e.g., [1]): specifically these sections have been used to form the training set (section 2-21) of 39,832 sentences (950,026 words), the test set (section 23) of 2,416 sentences (56,683 words) and the validation set (section 24) of 3,677 sentences (85,335 words). The entire dataset used for our experiments includes therefore 45,925 sentences for a total of 1,092,044 words. The average sentence length is 24 in a range of 1-141 (1-67 in the test set). The labels (tags) on the nodes of the parse trees can be divided into POS tags, or pre-terminal tags, and non-terminal tags: the first ones dominate a single lexical item and indicate the grammatical function of the item (ex. a noun or a verb) while the latter ones dominate sequences called phrases that can be made of pre-terminal and/or non-terminal tags. In the Penn Treebank the POS tags are 45, and the non-terminal tags are 26. Although the syntactic annotation schema provides a wide range of semantic and coindexing information, we have used only syntactic information. The incremental parser suffers from the problem of left recursion. Since a left recursive structure can be arbitrarily nested, we cannot predict the correct connection-path incrementally. There are a few practical solutions in the literature (see, e.g.,[12]), but in the current work we have resorted to an immediate approach which is extensively implemented in the Penn Treebank schema: namely we *flatten* the tree structure and avoid the left recursion issue altogether. Consider as an example the application of the flattening procedure to a local tree like 1 that produces as a result a tree like 2:

1. [$_{NP}$ [$_{NP}$ DT NN] PP]
2. [$_{NP}$ DT NN PP]

In this work we convert a sequence of words in a sequence of POS tags and proceed thereafter only with that. This choice was aimed at analyzing what kind of preferences can be expressed and learned in terms of pure syntactical information. In future development of this work it will be interesting to establish the increase in performance once we introduce lexical information. As a final remark note that in the current work we do not investigate the problem of POS tagging, i.e. attributing the correct part of speech to each lexical item and we assume this information available.

4 ANALYSIS

Basing our intuitions on the domain knowledge we hypothesise a certain number of structural characteristics of the incremental trees that are likely to have an influence over the generalization performance. The features that we investigate are the outdegree of specially informative nodes (anchor, root) and the average outdegree, the complexity of incremental trees and of connection path structures measured by the number of nodes,and the height, information about the number of the competing alternatives, and the word index in the sentence. The intuition behind trying to analyze the influence of the outdegree on the net's performance, is that backpropagating the error becomes a more difficult task as the number of possible sources amongst which to partition the error increases. The reason for analyzing the influence of the number of nodes or the height is that a greater height implies more steps in the propagation of the information and a bigger number of nodes implies a dependency on a greater number of possible configurations. In order to measure the correlation between these features and the net's error we run an analyses employing the Spearman Rank Correlation test over a randomly sampled sub-set of 200 pairs (error, feature). The test indicates a significant positive correlation for all features except for the root outdegree (Rs=0.017), and a stronger negative correlation between the frequency of the connection path and the error (Rs=-0.39). The most significant positive correlations are with the size of the connection path (Rs=0.33) and the forest size (Rs=0.31). We therefore analyze in greater details the nature of the influence of connection paths' frequency.

4.1 Accuracy and connection path frequency

We start by evaluating the prediction for the first-pass attachment if the decision were based on the connection path frequency information only. In Figure 1 we report a comparative test between the preference expressed by the net and the preference obtained by selecting the incremental tree with the most frequent connection path. The test is done on the standard set of 2,400 sentences. The figure reports the proportion of correct guesses out of the total number of words with respect to the ranking position of the guess, i.e. for $x = i$ we consider the cases when the system has ranked the correct element within the first i positions. On the y axis we report the proportion of correctly classified elements in respect of the total number of positive elements. From Figure 1 we can reliably say that the net bases its decisions on something more than the pure frequency of connection paths. The anova test was used to determine the influence of the log-transformed frequency[5] on the network's accuracy. We analyze the *true positive* elements, i.e. elements that have been preferred by the net and that are correct, and the *false positive* elements, i.e. elements that have been preferred but that are wrong. For the true positive the mean log-frequency of the connection path was 9.16 against a mean of 5.24 for the second best ranked alternative. The difference being highly significant on the random sample of 100 pairs. For the false positive dataset there was no significant difference in the mean of the log frequency (7.43 for the correct element vs. 7.22 of the predicted element, $F < 1$). Notice also that the overall mean is much higher for the true positives than the false positives. This could be the result of a more skewed distribution of the true positives with the correct alternative having a much higher frequency than the other alternatives. This seems to indicate that the net finds it more difficult to express a preference when it cannot draw information directly from the frequency distribution of the alternative connection paths. In the following we test if, and how much, the net expresses a preference beyond just using the absolute frequency of the connection paths.

4.2 Simplicity and frequency

The aim of this analysis is to characterize the features of the incremental trees that are correctly classified in respect of the trees that are preferred by the network but that turn out to be incorrect. The characterization will be expressed in terms of a statistically significant difference in the average values of the features that we are investigating. More specifically we will try to identify some consistent properties of the set of true positive elements that distinguish them from the second preferred element, and we will do the same with the false positive and the correct element [6]. Observing the distribution of the values taken by these features, we note that the forest size, the anchor distance from the root, the root outdegree, and the average number of nodes do not exhibit a normal distribution. For these features we use the Wilcoxon Matched-Pairs Signed-Ranks Test on a random sample of 200 pairs from the dataset for each feature. For all the other features the anova test is used, randomly sampling 100 pairs from the dataset for each feature. The conclusion that we can draw from this analysis is that trees which are simpler in various senses are preferred by the network when the correct alternative is slightly more complex. The simplicity can be expressed in terms of a shorter connection path, shorter trees, or connection paths or trees with fewer nodes, or with nodes with a smaller outdegree. There seems moreover to be no meaningful effect of the maximum outdegree on the net false positive error. We finally note how the differences within the whole incremental tree are much smaller than that between the connection paths indicating that these latter ones are the key element responsible for the discrimination between the correct element and the incorrect chosen element. Examining the simplicity preference we note that all the features are strongly correlated and that there could be an underlying factor that is the direct cause of the preference, namely the combinatorial effects of the atomic grammar elements, which implies that simpler connection paths are more frequent. As a direct consequence the correct incremental trees are themselves simpler because derived by joining simpler elements. It seems, though, that the frequency explanation cannot account for all the cases, in other words it represents a valid heuristic but it does not capture the overall complexity of the problem. To assess this point, we investigated the case where the network predicts correctly but where the correctly predicted alternative is not the most frequent. By selecting these cases we can judge the effect of structural factors controlling for the confounding influence of frequency. We extracted the sub-sample that corresponds to this latter case, which accounts for the 10% of the cases. The result indicates that in those cases where the net is correct but does not rely on the frequency information the elements chosen are in fact slightly more complex (greater number of nodes 31 vs 30.8, greater height 7.74 vs. 7.45), though the anchor has preferably a much lower outdegree (2.49 vs. 4.77) and the anchor depth is much higher (6.58 vs. 3.13). This result can be explained considering that the most frequent connection paths tend to have very common anchors, such as VPs, and that this bias the attachment point toward this less deep nodes making them having a greater outdegree. The results, on the other hand, confirm the hypothesis that frequency is very strongly correlated with complexity. To assess this in a more sound statistical way we run the Pearson Correlation test on a sample of 10000 pairs of connection paths' number of nodes vs. log(freq). We obtained a correlation of rho=-0.3291 (statistical significance $p < 0.001$) indicating that simpler connection paths are reliably more frequent. In the following we try to decompose and study the source of ambiguity of the problem in its two main components: the ambiguity on the attachment point and the ambiguity on which connection path to use.

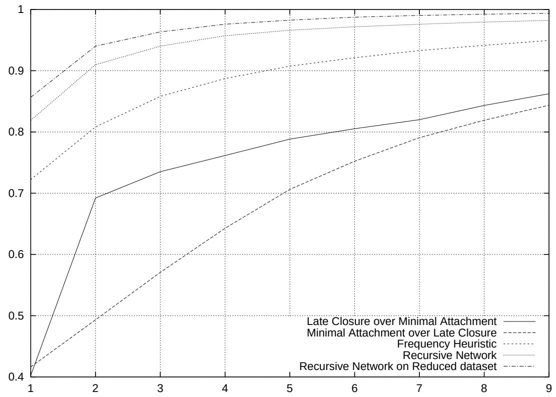

Figure 1. Comparison with psycholinguistic and frequency heuristics

4.3 Anchor attachment ambiguity

We tested whether the network uses some complex statistical analyses to disambiguate the attachment point of the anchor but resorts to frequency counts to choose the connection path. To verify this hypothesis we prepared a test set where the correct anchor is given and we collect the network's preferences on the remaining forest of competitors. In order to understand if the network uses only the frequency information of connection paths we counted the number of correct predictions where a connection path with lower frequency was preferred. Out of 88.5% correctly classified instances 3.8% were not frequency based. We also considered the wrong choices, evaluating how many times the network has made a mistake because it did prefer a more frequent connection path instead of the correct less fre-

[5] This is because the connection path frequency follows an exponential distribution.

[6] For more details see [4]

quent one. The result is that this is true in 66.3% of the cases where the net makes a mistake. Two thirds (2/3) of the mistakes can therefore be attributed to the net's preference for more frequent connection paths. The results obtained suggest a new strategy: we let the network choose the anchor using full information and then resort to the simple frequency of connection paths to choose among the alternatives. In order to verify this approach, we determine how frequently the net is capable of choosing a connection path that attaches to the correct anchor, even if the path itself is wrong. We collapsed all the connection paths that have the same anchor point into a single class and we counted how many times the correct connection path falls within that specific class. We then generalized this approach in order to take into consideration the second or third anchor chosen. We found that 91.54% of the times the network correctly predicts the anchor in the first choice, 97.11% of the times the anchor is predicted within the first 2 choices and 98.39% within the first 3. We then ranked the incremental trees preferred by the network as the first, second and third choice respectively. As ranking criterion we use the connection path's frequency and the net's preference. As a result we obtain for the frequency heuristic an accuracy respectively of 88.3%, 77.35% and 75.9% for the 1st, 2nd, 3rd choices, while adopting the preference expressed by the net we obtain an accuracy of 89.5%, 84.4%, 83.3%. The experiment suggests that the network is very successful in predicting the anchor and resorts to the frequency statistics for the prediction of the connection path. However, in comparison with the pure frequency estimation, the net allows more viable alternatives to compete in 2nd and 3rd position in the ranked list, indicating once again that the net is conditioning the statistics collected on some useful context present in the incremental tree. These results motivate a deeper analyses of the kind of statistics that the network is really employing.

4.4 Linguistic heuristics

Psycholinguistic studies suggest that the syntactic module of the human parser expresses some structural preferences among which the minimal attachment (MA) preference and the late closure (LC) preference ([7]). MA implies that humans tend to prefer simpler and shorter analyses (i.e. connection paths and incremetnal trees with fewer nodes). LC, instead, suggests that, a preference is expressed to connect the current analyses with recently processed material (i.e. low attachment anchor points are preferred). In Fig.1 we report the comparative test between the predictions expressed by the network and those expressed by the combined strategy LC-MA and MA-LC, where combining the strategies means that we resort to the second strategy for all those elements considered equal by the first strategy. The results show the effectiveness of these heuristics (see fig. 1); within the first two alternatives the LC-MA heuristic finds the correct element 70% of the times (while MA-LC 50% of the times). The network guesses correctly more than 82% of times with the first proposed element and more than 90% within the first two proposed elements. In order to test whether the network has learned to express preferences that mimic the analyzed heuristics or rather has found an orthogonal set of features we analyzed the prediction overlapping. We considered how many times the network first choice corresponds exactly to the first element ranked by each heuristic combination. The results indicate that the network resolves to the heuristics only half of the time (43.5% match between Net and LC-MA, 44.5% Net vs. MA-LC). If we allow the first or second choice of the network to match the first or second choice of the heuristics combination we find that between the net's preference and the LC then MA heuristic we have an agreement more than 78% of the times (61.5% for Net MA-LC). This allows us to infer that the network has learned to use the LC-MA preference, but that has also found other statistics to rely upon, which explains the 20% difference with respect to the LC-MA.

5 ENHANCEMENTS

5.1 Tree reduction

The experimental results reported in Section 4 have shown how the complexity of the incremental trees negatively affects the prediction performance. We would like to decrease this complexity (i.e. the number of nodes) without risking to disregard useful features. Intuitively not all the information of the incremental tree is significant for the disambiguation task. Specifically it can be argued that the knowledge of the internal composition of "closed" constituents, i.e. constituents that have been fully parsed, can be summarized by the non-terminal tag that immediately dominates the constituent. In other words, the knowledge that a deeply nested NP is made of a sequence of (DT NN) or rather a more complex (DT JJ NN NN) is not informative with respect to deciding how to attach a connection path. If this hypothesis is true it should be possible to eliminate a significant part of the nodes of the incremental tree without decreasing the discriminating power of the information that is left in the remaining nodes. We propose a reducing scheme where we keep all the nodes that dominate incomplete components plus all their children. Because of the incremental nature of the algorithm, it turns out that these nodes belong to the right frontier of the incremental tree. The procedure we are adopting turns out to be consistent with the notion of c-command. When we create Ti, we keep only the right frontier of Ti-1 and those nodes that c-command the right frontier itself. Preserving the nodes that c-command the nodes that are active (those that are potential anchors) is linguistically motivated in that it keeps the nodes that can exhibit a "linguistic influence" on each other. In Figure 2 we show the subset of retained nodes. In order to test the equivalence hypothesis we run an experiment with the following setting. The datasets are the standard training, validation and test sets where we have applied the reduction procedure. The network has 20 units in the recursive and output part. At each epoch the parameters of each net are saved. The performance in generalization of each net are measured against the validation set. The best performing parameters configuration is then tested on the test set. We report in figure 1 the comparison between the performance on the reduced dataset and the normal dataset. The results indicate that not only we have not eliminated relevant information, but we have helped the system eliminating potential sources of noise making the task somewhat simpler and allowing for a better generalization. To explain this behavior we

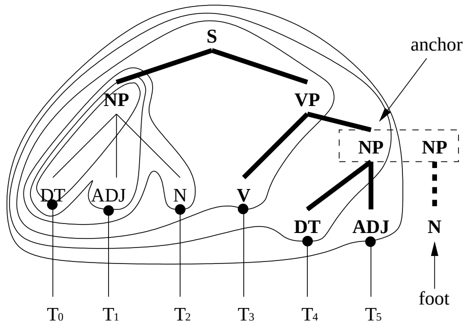

Figure 2. Sequence of incremental trees $T_0 \dots T_5$. In bold the result of the tree reduction procedure. In dashed lines the connection path.

can hypothesize the fact that the states that encode the information relative to what lays in deep (i.e. more distant from the root) nodes are “noisy” and confound higher (i.e. closer to the root) states.

5.2 Specialization on POS tags

When the learning domain can naturally be decomposed in a set of disjoint sub-domains, it is possible to specialize several learners on each sub-domain being confident that we are not decreasing the overall discriminating performance of the system. The domain in which we are operating lends itself to such a decomposition. For example, the knowledge acquired by the system in processing the attachment of verbs, is quite different from that used to attach articles or punctuation elements, i.e. in the two cases the features that are relevant for discriminating the correct incremental trees, differ. The knowledge of the domain suggests that certain attachment decisions will be harder than others. For example, the prepositional attachment is notoriously a hard problem, especially when full-lexical information is not used (as it is not in our case). We verify this hypothesis with an experiment that has the following setting. We divide the set of POS tags into 10 sub-sets where we collate “similar” tags, i.e. tags that have a similar grammatical function[7]. A special set contains all those tags that couldn’t be put in any other sub-set[8]. A network of 25 units, trained on the standard training set is then used on the standard test set. The prediction results are collected and partitioned in the appropriate subsets accordingly to which POS tag was involved in the attachment decision. We report the results in Table 1, where A1 is the best accuracy obtained using the method of Section 5.1 while in column Freq we report the fraction of the total dataset represented by each sub-set. The results indicate that the problem is harder in the case of adverbs and prepositions and easier for nouns, verbs and articles. We propose to enhance the overall performance letting single nets to concentrate on specific ambiguities, i.e. having a net being exposed only to attachment decisions involving, for example, adverbs or prepositions. The setting of the experiment is as described in Section 5.1. We report in table 1 the comparison between the performance of the specialized networks (column A2) and the unspecialized network (column A1) on the same dataset and the relative error reduction (column Err. red). The results indicate that we have an overall enhancement of the performance (15.3% error reduction) and that some categories greatly benefit from this approach. We believe that the reason is that the resources (i.e. areas in the state space) allocated for discriminating the less frequent classes (conjunctions, punctuation, adverbs) do not have to compete against the ones allocated for the most frequent cases (nouns, verbs).

Category	Freq	A1	A2	Rel. err red
Article	12.4	89.09	90.97	17.2
Preposition	12.6	64.26	68.19	11.0
Adjective	7.4	87.00	88.74	13.4
Verb	13.5	94.72	96.41	32.0
Noun	32.9	94.52	95.53	18.4
Possessive	2	97.99	97.13	-42.8
Adverb	4.2	53.46	55.88	5.2
Conjunction	2.3	70.41	76.28	19.8
Punctuation	11.7	75.29	80.84	22.5
Other	1	68.64	72.88	13.5
Total	100	84.82	87.14	15.3

Table 1. Specialization improvement

6 CONCLUSIONS

We have shown how the analyses of the preferences expressed by the recursive neural network allow to have a useful insight on the nature of the statistics information used by the system. We have found that the net bases its preferences to disambiguate the anchor attachment point on some complex information and mainly resorts to frequency to choose the correct connection path. Moreover, we have shown how the system prefers to attach simple structures to recently processed material, in a similar way to some heuristics proposed in the psycholinguistic literature, but that the incremental tree offers a richer context on which to condition the preferences which can be exploited by the proposed architecture. From the domain knowledge we have been able to use the above findings to propose a reduction scheme and a specialized architecture capable to enhance the overall prediction accuracy of the network. We believe that further improvements are achievable only introducing more information by lexicalizing the incremental grammar. Future work will focus on the use of the recursive neural network as an informant to guide an incremental parser.

REFERENCES

[1] M. J. Collins, ‘A new statistical parser based on bigram lexical dependencies’, in *Proceedings of the 34th annual meeting of the Association for Computational Linguistics*, (1996).

[2] F. Costa, P. Frasconi, V. Lombardo, and G. Soda, ‘Towards incremental parsing of natural language using recursive neural networks’, *Applied Intelligence*, (2002). In press.

[3] F. Costa, P. Frasconi, V. Lombardo, and P. Sturt, ‘An experience-based model of incremental parsing using dynamic grammars and recursive neural networks’, in *Proceedings of 6th Annual Conference on Architectures and Mechanisms for Language Processing*, (2000).

[4] F. Costa, P. Frasconi, V. Lombardo, P. Sturt, and G. Soda, ‘Improved ranking of structural preferences in incremental dynamic grammars’, (2002). In preparation.

[5] K. M. Eberhard, M. J. Spivey-Knowlton, J.C. Sedivy, and M. K. Tanenhaus, ‘Eye movements as a window into real-time spoken language comprehension in natural contexts’, *Journal of Psycholinguistic Research*, **24**, 409–436, (1995).

[6] P. Frasconi, M. Gori, and A. Sperduti, ‘A general framework for adaptive processing of data structures’, *IEEE Transactions on Neural Networks.*, **9**, 768–786, (1998).

[7] L. Frazier, *On comprehending sentences: Syntactic parsing strategies*, Ph.D. dissertation, University of Connecticut, Storrs, CT, 1978.

[8] P. C. R. Lane and J. B. Henderson, ‘Incremental syntactic parsing of natural language corpora with simple syncrony networks’, *IEEE Transactions on Knowledge and Data Engineering*, **13**(2), (2001).

[9] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, ‘Building a large annotated corpus of English: the Penn Treebank’, *Computational Linguistics*, **19**, 313–330, (1993).

[10] B. Roark, ‘Probabilistic top-down parsing and language modeling’, in *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pp. 249–276, (2001).

[11] P. Sturt, V. Lombardo, F. Costa, and P. Frasconi. A wide coverage model of first-pass preferences in human parsing. Paper presented at the 14th CUNY Sentence processing conference, Philadelphia, PA, March 2001.

[12] H. Thompson, M. Dixon, and J. Lamping, ‘Compose-reduce parsing’, in *Proceedings of the 29th Meeting of the Association for Computational Linguistics*, pp. 87–97, Berkley, California, (June 1991).

[7] For example all the tags MD VB VBD VBG VBN VBP VBZ are together under the category VERB.

[8] It includes POS tags that denote foreign words, exclamations, symbols, etc