PP-Attachment: A Committee Machine Approach

Martha A. Alegre
Departament de LSI
Universitat Politècnica de Catalunya
malegre@lsi.upc.es

Josep M. Sopena
Lab. de Neurocomputació
Universitat de Barcelona
pep@axon.psi.ub.es

Agusti Lloberas
Lab. de Neurocomputació
Universitat de Barcelona
agusti@axon.psi.ub.es

Abstract

In this paper we use various methods for multiple neural network combination in tasks of prepositional phrase attachment. Experiments with aggregation functions such as unweighted and weighted average, OWA operator, Choquet integral and stacked generalization demonstrate that combining multiple networks improve the estimation of each individual neural network. Using the Ratnaparkhi data set (the complete training set and the complete test set) we obtained an accuracy score of 86.08%. In spite of the high cost in computational time of neural net training, the response time in test mode is faster than others methods.

1 Introduction

Structural ambiguity is one of the most serious problems that Natural Language Processing (NLP) systems face. This ambiguity takes place because the syntactic information alone does not suffice to make an assignment decision. Constructions such as Prepositional Phrase (PP), coordination, or relative clauses are affected. An exhaustive study about the information needed to deal with this particular structural ambiguity has not been carried out as of yet; nevertheless, in the current literature we can find several proposals.

- In certain cases, it seems that the information needed to solve the attachment comes from the general context.

  (1.a) *John has a telescope.*
  (1.b) *He saw the girl with the telescope.*

  In this particular case, a correct attachment would require a model representing the situation in which the different entities are involved. If this were true for all of the cases, determining PP assignment would require highly complex computation.

- In some other cases, the information determining the PP attachment seems to be local. Some works [Woods et al, 1972], [Boguraev, 1979], [Marcus et al. 1993] suggested several strategies that based their

decision-making on the relationships existing between predicates and arguments– what [Katz and Fodor, 1963] called *selectional restrictions*. Cases belonging to this group seem to be easier to handle computationally than the former ones.

Regarding these different cases we can speak of two kinds of disambiguation mechanisms. One that can be called a low level mechanism which uses mainly information regarding *selectional restrictions* between predicates and arguments. This mechanism uses a local context in order to solve syntactic disambiguation: that which is constituted by the predicate and its arguments. The second mechanism uses higher level information such as situation models. If the low level mechanism does not solve the ambiguity, the high level mechanism, which would be activated later, should be able to do it. There are empirical data that seem to support the fact that human beings use these two mechanisms both for word sense disambiguation and syntactic disambiguation. For a review see [Sopena et al. 1998].

1.1 Local disambiguation

The low level disambiguation for the PP is one task that has been somewhat successfully treated using statistical methods. Not all of the methods use the *selectional restrictions* mechanism since they don't make use of semantic classes. We will use the term *local disambiguation* to encompass the methods based on *selectional restrictions* as well as those based on lexical association.

1.2 Selectional restrictions and PP-attachment

A system that correctly uses the information of semantic classes must first choose the correct sense of each word. If a hierarchy is used, an adequate level of abstraction must be determined. In Figure 1 it is shown how the level of abstraction can change depending on the verb. Considering the following examples :
(3.a) *Give access to documents*
(3.b) *Give a present to the driver*

In WordNet (WN) *give* has 27 senses, *driver* 4 senses and *present* 3 senses. With the co-occurrence of *give*, *driver*, *present*, the senses "give something to someone", "vehicle operator " and "gift" respectively are selected. The other senses not selected can be considered as noise. On the other hand, the adequate level of abstraction of *driver* is PERSON. The adequate level of abstraction of *present* is OBJECT.

(2.a) *To eat the strawberry with pleasure*

ENTITY
OBJECT
SUBSTANCE
FOOD ———— Adequate level of
GREEN GOODS abstraction
EDIBLE FRUIT of strawberry in (2.a)

(2.b) *To take a strawberry from the box.*

ENTITY
OBJECT
SUBSTANCE ——— Adequate level of
FOOD abstraction
GREEN GOODS of strawberry in (2.b)
EDIBLE FRUIT

Figure 1.

Most of the statistical methods that have used classes do not carry out a prior disambiguation of the words [Brill, Resnick 1994], [Ratnaparkhi et. al 1994] and others, nor do they determine the adequate level of abstraction. Some that *do* make the determination have a poor level of efficiency.

Table 1 shows the accuracy of the results reported in previous work. The worst results were obtained when only classes were used.

Stettina and Nagao used the Ratnaparkhi data set but they eliminated 3,224 4-tuples (15%) from the training set containing contradicting examples.

For reasons of complexity, the complete 4-tuple has not been considered simultaneously except in [Ratnaparkhi et. al 1994].

Classes of a given sense and classes of different senses of different words can have complex interactions and the preceding methods cannot take such interactions into account.

Neural networks (NNs) are appropriates in dealing with this complexity. A very impor-

Author	Best	Classes	Use of
			Ratnaparkhi set
Hindle and Rooth (1993)	80.0 %	No	No
Resnik and Hearst (1993)	83.9 %	WN	No
Resnik and Hearst (1993)	75.0 %	WN	No
Ratnaparkhi et al. (1994)	81.6 %	MIC	Yes
Brill and Resnik (1994)	81.8 %	WN	No
Collins and Brooks (1995)	84.5 %	No	Yes
Stettina and Nagao (1997)	88.0 %	WN	Yes
Sopena et al. (1998)	86.2 %	WN	No
Li and Abe (1998)	82.4 %	WN	No

Table 1: Test and accuracy results reported in previous works.

tant characteristic of NNs is their capacity to deal with multidimensional inputs. They need much fewer parameters to achieve the same result than traditional numerical methods. Recently [Barron, 1993] has shown that feedforward networks with one layer of sigmoidal nonlinearities achieve an integrated squared error of order $O(\frac{1}{n})$ for input spaces of dimension d, where n is the number of units of the network. Traditional methods (series expansions) with n terms can only achieve an integrated squared error of order $O((\frac{1}{n})2^d)$, for functions satisfying the same smoothness assumption. NNs are surprisingly advantageous in high dimensional inputs since the integrated squared error is independent of the input dimension. They can compute very complex statistical functions, they are model free, and compared to the current methods used by the statistical approach to NLP, NNs offer the possibility of dealing with a more complex (non-linear and multivariant) approach.

In the next section we describe a PP attachment disambiguation system based on neural networks that takes better advantage of the use of classes.

2 A neural network approach to PP-attachment

The use of classes is fundamental when working with neural networks. Using words alone without their classes in real texts, floods the memory capacity of a neural network. It is well known that the use of *words* creates huge probabilistic tables. In addition, the use of classes successfully deals with problems of invariance related to compositionality and binding that neural networks have [Sopena, 1996]. PP attachment can be considered as a classification problem were 4-tuples are classified in two classes: as to whether it is attached to the noun or to the verb [Sopena et al. 1998].

These classes are represented in the output units. When a local representation for classes is used (one class per unit) the output activation of each unit can be interpreted as the Bayesian posterior probability that the pattern in the input belongs to the class represented by this unit. In our case we have two units: one representing the class "attached to noun" and the other the class "attached to verb". The activitation of these units will represent the respective probability of attachment given the 4-tuple encoded in the input.

Given the set of words in the 4-tuple we have to determine a way to represent senses and semantic class information. Polysemy represents a problem when using word classes. In order to use class information, two different procedures are possible. The first one consists in presenting all the classes of each sense of each word serially. The second one consists in the simultaneous presentation of all the senses of all the words. In previous works we have found that parallel presentation improve results.

The parallel procedure has the advantage of detecting in the network classes that are related

to others within the same slot or among different slots.

Presenting all of the classes simultaneously (including verb classes) allows us to detect complex interactions among them (either the classes of a particular sense or the classes of different senses of a particular word) that cannot be detected in most of the methods used so far. We have been able to detect their existence in our studies on word sense disambiguation currently being carrying out. If we present simultaneously all the classes of all the senses of each word in the 4-tuple we will have a very complex input. A system capable of dealing with such an input would be able to select classes (and consequently senses) which are compatible with other ones.

Finally, and related to the above, most of the statistical methods used in Natural Language Processing are linear. Multilayer feedforward networks are non linear. One of the objectives of experiments is to see if introducing non-linearity improves the results.

2.1 Test and training data

We used the same data set (the complete training set and the complete test set) as [Ratnaparkhi et. al 1994] for purposes of comparison. In this data set the 4-tuples of the test and training sets were extracted from Penn Treebank Wall Street Journal [Marcus et al. 1993]. The test data consisted of 3,097 4-tuples with 20,801 4-tuples for the training data.

The following process was run over both test and training data:
- All numbers were replaced by the string "whole_number"
- All verbs and nouns were reduced to their morphological stem using WordNet.
- 275 nouns which had not been found in WordNet[Miller et. al.,1993] were replaced by synonyms using WordNet.
- The remaining nouns and verbs which had not been found in WordNet were left uncoded.
- Proper nouns were replaced by WordNet classes like "person", "business_organization", "social_group".
- Prepositions found fewer than 20 times either were not represented.

2.2 Codification

The input was divided into eight slots. The first four slots represented 'verb', 'n1', 'prep', and 'n2' respectively. In slots 'n1' and 'n2' each sense of the corresponding noun was encoded using all the classes within the IS-A branch of the WordNet hierarchy. This was done from the corresponding hierarchy root node to its bottom-most node. In the verb slot, the verb was encoded using the IS-A-WAY-OF branches. Each node in the hierarchy received a local encoding. There was a unit in the input for each node of the WordNet subset. This unit was ON if it represented a semantic class to which one of the senses of the encoded word belonged.

Using a local representation we needed a unit for each class-synset. The number class-synsets in WordNet is too large for a neural network. In order to reduce the number of input units we did not use WordNet directly, but constructed a new hierarchy (a subset of WordNet) including only the classes that corresponded to the words that belonged to the training and test sets.

A feedforward neural network can make good use of class information if there is a sufficient number of examples belonging to each class. For that reason we also counted the number of times the different semantic classes appeared in the training and test sets. The hierarchy was pruned taking these statistics into account. Given a threshold h, classes which appeared less than $h\%$ were not included. In all the experiments of this paper, we used tree cut thresholds of 2% . Regarding prepositions, only the 36 most frequent ones were represented (those found more than 20 times). For those, a local encoding was used. The rest of the prepositions were left uncoded.

The fifth slot represented the prepositions that the verb subcategorized. By representing the prepositions, [Sopena et al. 1998] had obtained improved results. The reason for this improvement being that English verbs with semantic similarity may take on different prepositions (for example, *accuse* with *of* and *blame* with *for*). Apart from semantic classes, verbs can also be

classified on the basis of the kind of prepositions they make use of.

The prepositions that the verbs subcategorize were initially extracted from COMLEX [Wolff et al., 1995]. Upon observation that COMLEX does not consider all the subcategorized prepositions, we complemented COMLEX with information extracted from training data. The prepositions of all the 4-tuples assigned to the verb were considered. The distinction between PP adjuncts and PP close-related were not available in the Ratnaparkhi data set. Therefore, we grouped the subcategorized prepositions by their verbs as well as those that govern PP adjuncts. Only the 36 most frequent prepositions were represented.

The sixth slot represented the prepositions that were governed by 'n1'. Again, only the 36 most frequent prepositions were represented. These prepositions were extracted from the 4-tuples of the training data whose attachments were to the noun.

The next slot represented 15 units for the lexicography verb files of WordNet. WordNet has a large number of verb root nodes, some of which are not frequent. Due to this fact, in some cases the pruning that was carried out on the tree made root nodes disappear. This lead to some of the verbs that belonged to this class not being coded. In order to avoid these cases, we used the names of the WordNet verb lexicographical files to add a new top level in the WordNet verb class hierarchy. Finally, in the last slot there are 2 units to indicate whether or not the N1 or N2 respectively were proper nouns .

Regarding the output, there were only two units representing whether the PP was attached to the verb or to the noun.

Feedforward networks with one hidden layer and full interconnectivity between layers were used in all the experiments. The networks were trained with the backpropagation learning algorithm. The activation function was the hyperbolic tangent function. The number of hidden units used was 0, 50 and 100. For all simulations the momentum was 0, and the initial weight range 0.1.

A validation set was constructed using 12,029 4-tuples extracted from the Brown Corpus.

In each run the networks were trained for 60 epochs storing the epoch weights with the smallest error regarding the validation set, as well as the weights of the 60th epoch (without the validation set).

3 Experiments

Table 2 shows the results of 24 training simulations obtained in the test data using 0, 50 and 100 hidden units respectively. We show the best results by the networks acting individually.

Methods	Best Results	60th epoch	Seconds
Perceptron	83.08 %	82.67 %	16
50 Hidden	85.18 %	84.21 %	44
100 Hidden	85.37%	84.50 %	78
Backed-Off	84.50%	-	230

Table 2: Results obtained with Backed-Off, 0, 50 and 100 hidden units and time in seconds to disambiguate 3,097 4-tuples.

In spite of the high cost in computational time of neural net training, the response time in test mode is up 3 times faster than Backed-Off model. This is shown in Table 2 where the time taken to disambiguate 3,097 4-tuples is given.

In this problem we had a high level of noise: on one hand the inadequate senses of each word in the 4-tuple. Words in English have a high number of senses thus, in the input, the level of noise (inadequate sense) can reach 5 times that of signal (correct sense). In addition, the Ratnaparkhi data set contains many errors, some of them due to errors originating from the Penn Treebank I. This level of noise deteriorates the generalizing capacity of the neural network.

There are many methods that permit a neural network to improve its capacity of generalization. For reasons of complexity, the size of the network that we are using places restrictions on the selection of the method. Of the methods that we are testing, committee machines allow us to improve results the most easily.

3.1 Experiments with Committees of networks:

The performance of a committee machine [Perrone, Cooper, 1993], [Perrone, 1994] and [Bishop C., 1995] can outperform that of the best single network used in isolation.

As [Kuncheva et al. 1998] points out, the process of combining multiple classifiers to achieve higher accuracy is given different names in the literature apart from *committee machines*: combination, classifier fusion, mixture of experts, consensus aggregation, classifier ensembles, etc. We have applied the following algorithms: average, weighted average, OWA operator, Choquet integral and stacked generalization.

3.1.1 Average :

Suppose we have a set of N trained network models $y_i(x)$ where $i = 1, ..., N$.

We can then write the mapping function of each network as the desired function $t(x)$ plus an error function [Bishop C., 1995]:

$$y_i(x) = t(x) + e_i(x)$$

The average sum-of-squares error for model $y_i(x)$ can be written as

$$E_i = E[(y_i(x) - t(x))^2] = E[e_i^2]$$

The output of the committee is the average of the outputs of the N networks that integrates the committee, in the form

$$y_{COM}(x) = \frac{1}{N}\sum_{i=1}^{N} y_i(x).$$

If we make the assumption that the errors $e_i(x)$ have zero mean and are uncorrelated, we have

$$E_{COM} = \frac{1}{N}E_{AV}$$

where E_{COM} is the average error made by the committee and E_{AV} is the average error made by the networks acting individually.

In general, the errors $e_i(x)$ are highly correlated but even then it is easy to show that [Bishop C., 1995]:

$$E_{COM} \leq E_{AV}$$

As some members of the committee will invariably give better results than others, it is of interest to give more weight to some of the members than to others taking the form:

$$y_{GEN}(x) = \sum_{i=1}^{N} \omega_i y_i(x)$$

here ω_i is based on the error of the validation and learning set.

3.1.2 The Ordered Weighted Averaging (OWA) Operators

If w is a weighting vector of dimension n, then a mapping $OWA_w : R^n \rightarrow R$ is an *Ordered Weighted Averaging (OWA) operator* of dimension n [Yager, 1993] :

$$OWA(y_1, \cdots, y_n) = \sum_{i=1}^{n} w_i y_\sigma(i)$$

where $\{\sigma(1), \cdots, \sigma(n)\}$ is a permutation of $\{1, \cdots, n\}$ such that $y_\sigma(i-1) \geq y_\sigma(i)$ for all $i = 2, \cdots, n$.

The OWA operator permits weighting the values in relation to their ordering.

Results are show in Tables 3, 4 and 5.

3.1.3 Choquet integral:

The fuzzy integral introduced by [Choquet G, 1954] and the associated fuzzy measures, provide a useful way for aggregation information. A fuzzy measure u defined on the measurable space (X,X) is a set function $u : X \rightarrow [0, 1]$ satisfying the following axioms:

$(i) u(\oslash) = 0,\ u(X) = 1$ (*boundary conditions*)

$(ii) A \subseteq B \rightarrow u(A) \leq u(B)$ (*monotonicity*)

(X,X,u) is said to be a fuzzy measurable space.

If u is a fuzzy measure on X, then the *Choquet integral* of a function $f : X \rightarrow R$ with respect to u is defined by:

$$\int f du = \sum_{i=1}^{n} (f(y_s(i)) - f(y_s(i-1)))u(As(i))$$

where $f(y_s(i))$ indicates that the indices have been permuted so that
$0 \leq f(y_s(1)) \leq \cdots \leq f(y_s(n)) \leq 1$,
$A_s(i) = y_s(i), \cdots, y_s(n)$ and $f(y_s(0)) = 0$

One characteristic property of Choquet integrals is monotonicity, i.e., increases of the input lead to higher integral values. Results are shown in Table 6 and Table 7.

Nets	Average	Weighted Average	OWA
6	84.92 %	85.34 %	85.18 %
12	85.18 %	85.63 %	85.28 %
18	85.21 %	85.89 %	85.60 %
24	85.31 %	85.76 %	85.63 %

Table 3: Results Committee machines. 50 hidden layers

Nets	Average	Weighted Average	OWA
6	85.53 %	85.53 %	85.53 %
12	85.34 %	85.53 %	85.76 %
18	85.41 %	85.73 %	85.89 %
24	85.76 %	85.92 %	86.02 %

Table 4: Results Committee machines. 100 hidden layers

Nets	Average	Weighted Average	OWA
6	84.98 %	85.11 %	84.98 %
12	84.85 %	85.15 %	85.02 %
18	84.89 %	85.28 %	85.24 %
24	84.92 %	85.24 %	85.41 %

Table 5: Results Committee machines in 60th epoch. 100 hidden layers

Net 1	Net 2	Net 3	Choquet
84.50 %	83.34 %	83.60 %	84.92 %
84.50 %	84.24 %	84.18 %	85.02 %
84.82 %	85.37 %	84.76 %	85.66 %
84.79 %	84.57 %	84.57 %	85.24 %

Table 6: Results Choquet integral.

3.2 Stacked generalization:

[Wolpert, 1992] provides one way of combining trained networks which partitions the data set in order to find an overall system which usually improves generalization. The idea is to train the level-0 networks first and then examine their behavior when generalizing. This provides a new training set which is used to train the level-1 network. The inputs consist of the outputs of all the level-0 networks, and the target value is the corresponding target value from the original full data set. Our experiments using this method did not give improved results (85.35%).

Net 1	Net 2	Net 3	Net 4	Choquet
84.24 %	84.05 %	84.50 %	83.66 %	85.28 %
84.18 %	84.11 %	84.50 %	84.24 %	85.60 %
85.37 %	84.76 %	84.79 %	84.63 %	85.79 %
84.79 %	84.76 %	85.37 %	84.82 %	86.08 %

Table 7: Results Choquet integral.

4 Conclusions

Neural networks have been shown to be very successful in tasks such as pattern recognition or prediction in many different applications of business, biomedicine, engineering, astronomy, high energy physics, etc. Their results are similar and often better than those of alternative models. The benefits of neural networks are well known as was explained above. Unfortunately neural networks have not been very successful in the domain of Natural Language Processing. However, our system has obtained better results than any that have been published to date using the complete Ratnaparkhi data set. We also obtained excellent results in word sense disambiguation [Moliner, 1998]. Our success can be attributed to two things: on one hand, the use of semantic classes is fundamental to keep from flooding the network's memory. In other hand, the use of canonic thematic structures. Finally, improvement on the generalization is an area in permanent development in the field of neural networks. We are developing new methods of generalization

which will allow us to improve our results even more. Provisional results place us in the environment of 88% with Ratnaparkhi's data set.

References

[Barron, 1993] Universal Approximation Bounds for Superposition of a Sigmoidal Function. *IEEE Transactions on Information Theory*, 39:930-945

[Bishop C., 1995] Neural Networks for pattern recognition.

[Boguraev, 1979] *Automatic resolution of linguistic ambiguities.* Ph.D. Computer Laboratory, University of Cambridge.

[Brill, Resnick 1994] A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation. *In Proceedings of the Fifteenth International Conferences on Computational Linguistics (COLING-94).*

[Choquet G, 1954] Theory of capacities. Annales de L'Institut Fourier, 5, 1953-54, pp. 131-295

[Katz and Fodor, 1963] . The Structure of Semantic Theory. *Language, 39*: 170-210.

[Kuncheva et al. 1998] On Combining Multiple Classifiers by Fuzzy Templates, *Proc. NAFIPS'98. Pensacola. Florida,* pp. 193-197.

[Marcus et al. 1993] Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*,19:313-330

[Miller et. al.,1993] Introduction to WordNet: An Online Lexical Database. Anonymous FTP, internet:clarity.princeton.edu.

[Moliner, 1998] Un enfoque neuronal de la desambiguacion del sentido de las palabras (A Neural Network approach to word sense disambiguation). Technical University of Catalonia. LSI Dept. Ph.D. Thesis. 1998.

[Perrone, 1994] General averaging results for convex optimization. In M. C. mozer et al. (Eds.), *Proceedings 1993 Connectionist Models Summer School,* pp. 364-371. Hillsdale, NJ: Lawrence Erlbaum.

[Perrone, Cooper, 1993] When networks disagree: ensemble methods for hybrid neural networks. In R. J. Mammone (Ed.), *Artificial Neural Networks for Speech and Vision,*, pp. 126-142. London: Chapman & Hall.

[Ratnaparkhi et. al 1994] A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the ARPA Workshop on Human Language Technology.*

[Sopena, 1996] Word sense disambiguation: a neural network approach. Technical report 3-96. Laboratori de Neurocomputacio. University of Barcelona.

[Sopena et al. 1998] A Connectionist Approach to Prepositional Phrase Attachment for Real World Texts. In COLING-ACL'98. pp. 1233-1237

[Wolff et al., 1995] *Comlex Word Classes.* C.S. Dept., New York U., Feb. prepared for the Linguistic Data Consortium, U. Pennsylvania.

[Wolpert, 1992] Stacked generalization. *Neural Networks* 5 (2), pp. 241-259.

[Woods et al, 1972] *The Lunar Sciences Natural Language Information System: Final report* Report 2378, Bolt, Beranek and Newman,Inc., Cambridge, MA.

[Yager, 1993] Families of OWA operators, *Fuzzy Sets and Systems*, 59 pp. 125-148.