Memory-Based Shallow Parsing

Walter Daelemans, Sabine Buchholz, Jorn Veenstra
ILK, Tilburg University, PO-box 90153, NL 5000 LE Tilburg
[walter,buchholz,veenstra@kub.nl]

cs.CL/9906005 2 Jun 1999

Abstract

We present a memory-based learning (MBL) approach to shallow parsing in which POS tagging, chunking, and identification of syntactic relations are formulated as memory-based modules. The experiments reported in this paper show competitive results, the $F_{\beta=1}$ for the Wall Street Journal (WSJ) treebank is: 93.8% for NP chunking, 94.7% for VP chunking, 77.1% for subject detection and 79.0% for object detection.

Introduction

Recently, there has been an increased interest in approaches to automatically learning to recognize shallow linguistic patterns in text [Ramshaw and Marcus, 1995, Vilain and Day, 1996, Argamon *et al.*, 1998, Buchholz, 1998, Cardie and Pierce, 1998, Veenstra, 1998, Daelemans *et al.*, 1999a]. Shallow parsing is an important component of most text analysis systems in applications such as information extraction and summary generation. It includes discovering the main constituents of sentences (NPs, VPs, PPs) and their heads, and determining syntactic relationships like subject, object, adjunct relations between verbs and heads of other constituents.

Memory-Based Learning (MBL) shares with other statistical and learning techniques the advantages of avoiding the need for manual definition of patterns (common practice is to use hand-crafted regular expressions), and of being reusable for different corpora and sublanguages. The unique property of memory-based approaches which sets them apart from other learning methods is the fact that they are *lazy learners*: they keep all training data available for extrapolation. All other statistical and machine learning methods are *eager* (or *greedy*) learners: They abstract knowledge structures or probability distributions from the training data, forget the individual training instances, and extrapolate from the induced structures. Lazy learning techniques have been shown to achieve higher accuracy than eager methods for many language processing tasks. A reason for this is the intricate interaction between regularities, subregularities and exceptions in most language data, and the related problem for learners of distinguishing noise from exceptions. Eager learning techniques abstract from what they consider noise (hapaxes, low-frequency events, non-typical events) whereas lazy learning techniques keep all data available, including exceptions which may sometimes be productive. For a detailed analysis of this issue, see [Daelemans *et al.*, 1999a]. Moreover, the automatic feature weighting in the similarity metric of a memory-based learner makes the approach well-suited for domains with large numbers of features from heterogeneous sources, as it embodies a smoothing-by-similarity method when data is sparse [Zavrel and Daelemans, 1997].

In this paper, we will provide a empirical evaluation of the MBL approach to syntactic analysis on a number of shallow pattern learning tasks: NP chunking, VP chunking, and the assignment of subject-verb and object-verb relations. The approach is evaluated by cross-validation on the WSJ treebank corpus [Marcus *et al.*, 1993]. We compare the approach qualitatively and as far as possible quantitatively with other approaches.

Memory-Based Shallow Syntactic Analysis

Memory-Based Learning (MBL) is a classification-based, supervised learning approach: a memory-based learning algorithm constructs a classifier for a task by storing a set of examples. Each example associates a feature vector (the problem description) with one of a finite number of classes (the solution). Given a new feature vector, the classifier extrapolates its class from those of the most similar feature vectors in memory. The metric defining similarity can be automatically adapted to the task at hand.

In our approach to memory-based syntactic pattern recognition, we carve up the syntactic analysis process into a number of such classification tasks with input vectors representing a focus item and a dynamically selected surrounding context. As in Natural Language Processing problems in general [Daelemans, 1995], these classification tasks can be segmentation tasks (e.g. decide whether a focus word or tag is the start or end of an NP) or disambiguation tasks (e.g. decide whether a chunk is the subject NP, the object NP or neither). Output of some memory-based modules (e.g. a tagger or a chunker) is used as input by other memory-based modules (e.g. syntactic relation assignment).

Similar cascading ideas have been explored in other approaches to text analysis: e.g. finite state partial parsing [Abney, 1996, Grefenstette, 1996], statistical decision tree parsing [Magerman, 1994], maximum entropy parsing [Ratnaparkhi, 1997], and memory-based learning [Cardie, 1994, Daelemans *et al.*, 1996].

Algorithms and Implementation

For our experiments we have used TiMBL[1], an MBL software package developed in our group [Daelemans *et al.*, 1999b]. We used the following variants of MBL:

- IB1-IG: The distance between a test item and each memory item is defined as the number of features for which they have a different value (overlap metric). Since in most cases not all features are equally relevant for solving the task, the algorithm uses information gain (an information-theoretic notion measuring the reduction of uncertainty about the class to be predicted when knowing the value of a feature) to weight the cost of a feature value mismatch during comparison. Then the class of the most similar training item is predicted to be the class of the test item. Classification speed is linear to the number of training instances times the number of features.

- IGTree: IB1-IG is expensive in basic memory and processing requirements. With IGTree, an oblivious decision tree is created with features as tests, and ordered according to information gain of features, as a heuristic approximation of the computationally more expensive pure MBL variants. Classification speed is linear to the number of features times the average branching factor in the tree, which is less than or equal to the average number of values per feature.

For more references and information about these algorithms we refer to [Daelemans *et al.*, 1999b, Daelemans *et al.*, 1999a]. In [Daelemans *et al.*, 1996] both algorithms are explained in detail in the context of MBT, a memory-based POS tagger, which we presuppose as an available module in this paper. In the remainder of this paper, we discuss results on the different tasks in section *Experiments*, and compare our approach to alternative learning methods in section *Discussion and Related Research.*

[1] TiMBL is available from: http://ilk.kub.nl/

Experiments

We carried out two series of experiments. In the first we evaluated a memory-based NP and VP chunker, in the second we used this chunker for memory-based subject/object detection.

To evaluate the performance of our trained memory-based classifiers, we will use four measures: accuracy (the percentage of correctly predicted output classes), precision (the percentage of predicted chunks or subject- or object-verb pairs that is correct), recall (the percentage of chunks or subject- or object-verb pairs to be predicted that is found), and F_β [C.J.van Rijsbergen, 1979], which is given by $\frac{(\beta^2+1).prec.rec}{\beta^2.prec+rec}$, with $\beta = 1$. See below for an example.

For the chunking tasks, we evaluated the algorithms by cross-validation on all 25 partitions of the WSJ treebank. Each partition in turn was selected as a test set, and the algorithms trained on the remaining partitions. Average precision and recall on the 25 partitions will be reported for both the IB1-IG and IGTREE variants of MBL. For the subject/object detection task, we used 10-fold cross-validation on treebank partitions 00–09. In section *Related Research* we will further evaluate our chunkers and subject/object detectors.

Chunking

Following [Ramshaw and Marcus, 1995] we defined chunking as a tagging task, each word in a sentence is assigned a tag which indicates whether this word is inside or outside a chunk. We used as tagset:

I_NP inside a baseNP.

O outside a baseNP or a baseVP.

B_NP inside a baseNP, but the preceding word is in another baseNP.

I_VP and B_VP are used in a similar fashion.

Since baseNPs and baseVPs are non-overlapping and non-recursive these five tags suffice to unambiguously chunk a sentence. For example, the sentence:

> $[_{NP}$ Pierre Vinken $_{NP}]$, $[_{NP}$ 61 years $_{NP}]$ old , $[_{VP}$ will join $_{VP}]$ $[_{NP}$ the board $_{NP}]$ as $[_{NP}$ a nonexecutive director $_{NP}]$ $[_{NP}$ Nov. 29 $_{NP}]$.

should be tagged as:

Methods	context	accuracy	precision	recall	$F_{\beta=1}$
	NPs				
IGTree	2-1	97.5	91.8	93.1	92.4
IB1-IG	2-1	98.0	93.7	94.0	93.8
baseline words	0	92.9	76.2	79.7	77.9
baseline POS	0	94.7	79.5	82.4	80.9
	VPs				
IGTree	2-1	99.0	93.0	94.2	93.6
IB1-IG	2-1	99.2	94.0	95.5	94.7
baseline words	0	95.5	67.5	73.4	70.3
baseline POS	0	97.3	74.7	87.7	81.2

Table 1: Overview of the NP/VP chunking scores of 25-fold cross-validation on the WSJ using IB1-IG with a context of two words and POS right and one left, and of using IGTree with the same context. The baseline scores are computed with IGTree using only the focus POS tag or the focus word

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	Class
Weight	39	40	4	3	2	10	12	18	29	18	31	13	24	
Inst.1	-1	0	0	seen	VBN	-	-	-	-	sisters	PRP$	seen	VBN	S
Inst.2	1	0	0	seen	VBN	sisters	PRP$	seen	VBN	man	NN	lately	RB	O
Inst.3	2	0	0	seen	VNB	seen	VBN	man	NN	lately	RB	.	.	-

Table 2: Some sample instances for the subject/object detection task. The second row shows the relative weight of the features (truncated and multiplied by 100; from one of the 10 cross-validation experiments). Thus the order of importance of the features is: 2, 1, 11, 9, 13, 10, 8, 12, 7, 6, 3, 4, 5.

Pierre$_{I_NP}$ Vinken$_{I_NP}$,$_{O}$ 61$_{I_NP}$ years$_{I_NP}$ old$_{O}$,$_{O}$ will$_{I_VP}$ join$_{I_VP}$ the$_{I_NP}$ board$_{I_NP}$ as$_{O}$ a$_{I_NP}$ nonexecutive$_{I_NP}$ director$_{I_NP}$ Nov.$_{B\_NP}$ 29$_{I_NP}$.$_{O}$

Suppose that our classifier erroneously tagged *director* as *B_NP* instead of *I_NP*, but classified the rest correctly. Accuracy would then be $\frac{17}{18} = 0.94$. The resulting chunks would be *[$_{NP}$ a nonexecutive $_{NP}$]* *[$_{NP}$ director $_{NP}$]* instead of *[$_{NP}$ a nonexecutive director $_{NP}$]* (the other chunks being the same as above). Then out of the seven predicted chunks, five are correct (precision= $\frac{5}{7}$ = 71.4%) and from the six chunks that were to be found, five were indeed found (recall= $\frac{5}{6}$ = 83.3%). $F_{\beta=1}$ is 76.9%.

The features for the experiments are the word form and the POS tag (as provided by the WSJ treebank) of the two words to the left, the focus word, and one word to the right. For the results see Table 1.

The baseline for these experiments is computed with IB1-IG, with as only feature: i) the focus word, and ii) the focus POS tag.

The results of the chunking experiments show that accurate chunking is possible, with $F_{\beta=1}$ values around 94%.

Subject/Object Detection

Finding a subject or object (or any other relation of a constituent to a verb) is defined in our classification-based approach as a mapping from a pair of words (the verb and the head of the constituent) and a representation of its context to a class describing the type of relation (e.g. subject, object, or neither). A verb can have a subject or object relation to more than one word in case of NP coordination, and a word can be the subject of more than one verb in case of VP coordination.

Data Format

In our representation, the tagged and chunked sentence

> **[NP** *My*/PRP$ *sisters*/NNS **NP] [VP** *have*/VBP *not*/RB *seen*/VBN **VP] [NP** *the*/DT *old*/JJ *man*/NN **NP]** *lately*/RB ./.

will result in the instances in Table 2.

Classes are *S(ubject)*, *O(bject)* or "-" (for anything else). Features are:

1 the distance from the verb to the head (a chunk just counts for one word; a negative distance means that the head is to the left of the verb),

	Together				Subjects			Objects		
# relations	51629				32755			18874		
Method	acc.	prec.	rec.	$F_{\beta=1}$	prec.	rec.	$F_{\beta=1}$	prec.	rec.	$F_{\beta=1}$
Random baseline		3.9	3.9	3.9	4.5	4.5	4.5	2.7	2.5	2.6
Heuristic baseline		65.9	66.5	66.2	69.3	61.6	65.2	61.6	75.1	67.7
IGTree	96.9	79.5	73.2	76.2	80.9	71.4	75.8	77.2	76.4	76.8
IB1-IG	96.6	74.4	**76.9**	75.6	76.2	**76.9**	76.5	71.5	**76.7**	74.0
IGTree & IB1-IG unanimous	97.4	**89.8**	68.6	**77.8**	**89.7**	67.6	**77.1**	**89.8**	70.4	**79.0**

Table 3: Results of the 10-fold cross validation experiment on the subject–verb/object–verb relations data. We trained one classifier to detect subjects as well as objects. Its performance can be found in the column *Together*. For expository reasons, we also mention how well this classifier performs when computing precision and recall for subjects and objects separately.

2 the number of other baseVPs between the verb and the head (in the current setting, this can maximally be one),

3 the number of commas between the verb and the head,

4 the verb, and

5 its POS tag,

6–9 the two left context words/chunks of the head, represented by the word and its POS

10–11 the head itself, and

12–13 its right context word/chunk.

Features one to three are numeric features. This property can only be exploited by IB1-IG. IGTree treats them as symbolic. We also tried four additional features that indicate the sort of chunk (NP, VP or none) of the head and the three context elements respectively. These features did not improve performance, presumably because this information is mostly inferrable from the POS tag.

To find subjects and objects in a test sentence, the sentence is first POS tagged (with the Memory-Based Tagger MBT) and chunked (see section *Experiments: Chunking*). Subsequently, all chunks are reduced to their heads.[2]

Then an instance is constructed for every pair of a baseVP and another word/chunk head provided they are not too distant from each other in the sentence. A crucial point here is the definition of "not too distant". If our definition is too strict, we might exclude too many actual subject-verb or object-verb pairs, which will result in low recall. If the definition is too broad, we will get very large training and test sets. This slows down learning and might even have a negative effect on precision because the learner is confronted with too much "noise". Note further that defining distance purely as the number of intervening words or chunks is not fully satisfactory as this does not take clause structure into account. As one clause normally contains one baseVP, we developped the idea of counting intervening baseVPs. Counts on the treebank showed that less than 1% of the subjects and objects are separated from their verbs by more than one other baseVP. We therefore construct an instance for every pair of a baseVP and another word/chunk head if they have not more than one other baseVP in between them.[3]

These instances are classified by the memory-based learner. For the training material, the POS tags and chunks from the treebank are used directly. Also, subject-verb and object-verb relations are extracted to yield the class values.

Results and discussion The results in Table 3 show that finding (unrestricted) subjects and objects is a hard task. The baseline of classifying instances at random (using only the probability distribution of the classes) is about 4%. Using the simple heuristic of classifying each (pro)noun directly in front of resp. after the verb as S resp. O yields a much higher baseline of about 66%. Obviously, these are the easy cases. IGTree,

[2] By definition, the head is the rightmost word of a baseNP or baseVP.

[3] The following sentence shows a subject-verb pair (in bold) with one intervening baseVP (in italics):
[$_{NP}$ The **plant** $_{NP}$], [$_{NP}$ which $_{NP}$] [$_{VP}$ *is owned* $_{VP}$] by [$_{NP}$ Hollingsworth & Vose Co. $_{NP}$] , [$_{VP}$ **was** $_{VP}$] under [$_{NP}$ contract $_{NP}$] with [$_{NP}$ Lorillard $_{NP}$] [$_{VP}$ to make $_{VP}$] [$_{NP}$ the cigarette filters $_{NP}$] .
The next example illustrates the same for an object-verb pair:
Along [$_{NP}$ the way $_{NP}$] , [$_{NP}$ he $_{NP}$] [$_{VP}$ **meets** $_{VP}$] [$_{NP}$ a solicitous Christian chauffeur $_{NP}$] [$_{NP}$ who $_{NP}$] [$_{VP}$ *offers* $_{VP}$] [$_{NP}$ the hero $_{NP}$] [$_{NP}$ God $_{NP}$] [$_{NP}$'s phone number $_{NP}$] ; and [$_{NP}$ the Sheep **Man** $_{NP}$] , [$_{NP}$ a sweet, rough-hewn figure $_{NP}$] [$_{NP}$ who $_{NP}$] [$_{VP}$ wears $_{VP}$] – [$_{NP}$ what else $_{NP}$] – [$_{NP}$ a sheepskin $_{NP}$] .

Method	Tagger	accuracy	precision	recall	$F_{\beta=1}$
A,D&K	Brill	–	91.6	91.6	91.6
R&M	Brill	97.4	92.3	91.8	92.0
C&P	Brill	–	90.7	91.1	90.9
IB1-IG	Brill	97.2	91.5	91.3	91.4
IB1-IG	MBT	97.3	91.6	91.5	91.6
IB1-IG	WSJ	97.6	92.2	92.5	92.3
IB1-IG,POSonly	WSJ	96.9	90.3	90.1	90.2

Table 4: Comparison of MBL and MBSL on same dataset of several classifiers, the experiments with IB1-IG are all carried out with a context of five words and POS left and three right

which is the better overall MBL algorithm on this task, scores 10% above this baseline, i.e. 76.2%. The difference in accuracy between IGTree and IB1-IG is only 0.3%. In terms of F-values, IB1-IG is better for finding subjects, whereas IGTree is better for objects. We also note that IGTree always yields a higher precision than recall, whereas IB1-IG does the opposite.

IGTree is thus more "cautious" than IB1-IG. Presumably, this is due to the word-valued features. Many test instances contain a word not occurring in the training instances (in that feature). In that case, search in the IGTree is stopped and the default class for that node is used. As the "-" class is more than ten times more frequent than the other two classes, there is a high chance that this default is indeed the "-" class, which is always the "cautious" choice. IB1-IG, on the other hand, will not stop on encountering an unseen word, but will go on comparing the rest of the features, which might still opt for a non-"-" class. The differences in precision and recall surely are a topic for further research. So far, this observation led us to combine both algorithms by classifying an instance as S resp. O only if both algorithms agreed and as "-" otherwise. The combination yields higher precision at the cost of recall, but the overall effect is certainly positive ($F_{\beta=1} = 77.8\%$).

Discussion and Related Research

In [Argamon *et al.*, 1998], an alternative approach to memory-based learning of shallow patterns, memory-based sequence learning (MBSL), is proposed. In this approach, tasks such as base NP chunking and subject detection are formulated as separate bracketing tasks, with as input the POS tags of a sentence. For every input sentence, all possible bracketings in context (situated contexts) are hypothesised and the highest scoring ones are used for generating a bracketed output sentence. The score of a situated hypothesis depends on the scores of the tiles which are part of it and the degree to which they cover the hypothesis. A tile is defined as a substring of the situated hypothesis containing a bracket, and the score of a tile depends on the number of times it is found in the training material divided by the total number of times the string of tags occurs (i.e. including occurrences with another or no bracket). The approach is memory-based because all training data is kept available. Similar algorithms have been proposed for grapheme-to-phoneme conversion by [Dedina and Nusbaum, 1991], and [Yvon, 1996], and the approach could be seen as a linear algorithmic simplification of the DOP memory-based approach for full parsing [Bod, 1995]. In the remainder of this section, we show that an empirical comparison of our computationally simpler MBL approach to MBSL on their data for NP chunking, subject, and object detection reveals comparable accuracies.

Chunking

For NP chunking, [Argamon *et al.*, 1998] used data extracted from section 15-18 of the WSJ as a fixed train set and section 20 as a fixed test set, the same data as [Ramshaw and Marcus, 1995]. To find the optimal setting of learning algorithms and feature construction we used 10-fold cross validation on section 15; we found IB1-IG with a context of five words and POS-tags to the left and three to the right as a good parameter setting for the chunking task; we used this setting as the default setting for our experiments. For an overview of the results see Table 4. Since part of the chunking errors could be caused by POS errors, we also compared the same baseNP chunker on the same corpus tagged with i) the Brill tagger as used in [Ramshaw and Marcus, 1995], ii) the Memory-Based Tagger (MBT) as described in [Daelemans *et al.*, 1996]. We also present the results of [Argamon *et al.*, 1998], [Ramshaw and Marcus, 1995] and [Cardie and Pierce, 1998] in Table 4. The latter two use a transformation-based error-driven learning method [Brill, 1992]. In [Ramshaw and Marcus, 1995], the method is used for NP chunking, and in

	Subjects			Objects		
# subsequences	3044			1626		
Method	prec.	rec.	$F_{\beta=1}$	prec.	rec.	$F_{\beta=1}$
A,D&K	88.6	84.5	**86.5**	77.1	89.8	83.0
IGTree	79.9	71.7	75.6	84.4	85.8	85.1
IB1-IG	84.7	81.6	83.1	87.3	85.8	**86.5**
IB1-IG POS only	83.5	77.9	80.6	76.1	83.3	79.6
IB1-IG without chunks	29.2	24.4	26.6	85.0	18.5	30.4
IB1-IG with treebank chunks	89.4	88.6	89.0	91.9	91.3	91.6

Table 5: Comparison of MBL and MBSL on subject/object detection as formulated by Argamon et al.

[Cardie and Pierce, 1998] the approach is indirectly used to evaluate corpus-extracted NP chunking rules.

As [Argamon *et al.*, 1998] used only POS information for their MBSL chunker, we also experimented with that option (POSonly in the Table). Results show that adding words as information provides useful information for MBL (see Table 4).

Subject/object detection

For subject/object detection, we trained our algorithm on section 01–09 of the WSJ and tested on Argamon et al.'s test data (section 00). We also used the treebank POS tags instead of MBT. For comparability, we performed two separate learning experiments. The verb windows are defined as reaching only to the left (up to one intervening baseVP) in the subject experiment and only to the right (with no intervening baseVP) in the object experiment. The relational output of MBL is converted to the sequence format used by MBSL. The conversion program first selects one relation in case of coordinated or nested relations. For objects, the actual conversion is trivial: The V–O sequence extends from the verb up to the head (*seen the old man* for the example sentence on page 3). In the case of subjects, the S–V sequence extends from the beginning of the baseNP of the head up to the first non-modal verb in the baseVP (*My sisters have*). The program also uses filters to model some restrictions of the patterns that Argamon et al. used for data extraction. They extracted e.g. only objects that immediately follow the verb.

The results in Table 5 show that highly comparable results can be obtained with MBL on the (impoverished) definition of the subject-object task. IB1-IG as well as IGTree are better than MBSL on the object data. They are however worse on the subject data. Two factors may have influenced this result. Firstly, more than 17% of the precision errors of IB1-IG concern cases in which the word proposed by the algorithm is indeed the subject according to the treebank, but the corresponding sequence is not included in Argamon et al.'s test data due to their restricted extraction patterns. Secondly, there are cases for which MBL correctly found the head of the subject, but the conversion results in an incorrect sequence. These are sentences like *"All [NP the man NP] [NP 's friends NP] came."* in which *all* is part of the subject while not being part of any baseNP.

Apart from using a different algorithm, the MBL experiments also exploit more information in the training data than MBSL does. Ignoring lexical information in chunking and subject/object detection decreased the $F_{\beta=1}$ value by 2.5% for subjects and 6.9% for objects. The bigger influence for objects may be due to verbs that take a predicative object instead of a direct one. Knowing the lexical form of the verb helps to make this distinction. In addition, time expressions like "(it rained) last week" can be distinguished from direct objects on the basis of the head noun. Not chunking the text before trying to find subjects and objects decreases F-values by more than 50%. Using the "perfect" chunks of the treebank, on the other hand, increases F by 5.9% for subjects and 5.1% for objects. These figures show how crucial the chunking step is for the succes of our method.

General

Clear advantages of MBL are its efficiency (especially when using IGTree), the ease with which information apart from POS tags can be added to the input (e.g. word information, morphological information, wordnet tags, chunk information for subject and object detection), and the fact that NP and VP chunking and different types of relation tagging can be achieved in one classification pass. It is unclear how MBSL could be extended to incorporate other sources of information apart from POS tags, and what the effect would be on performance. More limitations of MBSL are that it cannot find nested sequences, which nevertheless occur frequently in tasks such as subject identification[4], and that it does not mark heads.

[4] e.g. [SV John, who [SV I like SV], is SV] angry.

Conclusion

We have developed and empirically tested a memory-based learning (MBL) approach to shallow parsing in which POS tagging, chunking, and identification of syntactic relations are formulated as memory-based modules. A learning approach to shallow parsing allows for fast development of modules with high coverage, robustness, and adaptability to different sublanguages. The memory-based algorithms we used (IB1-IG and IGTree) are simple and efficient supervised learning algorithms. Our approach was evaluated on NP and VP chunking, and subject/object detection (using output from the chunker). $F_{\beta=1}$ scores are 93.8% for NP chunking, 94.7% for VP chunking, 77.1% for subject detection and 79.0% for object detection. The accuracy and efficiency of the approach are encouraging (no optimisation or post-processing of any kind was used yet), and comparable to or better than state-of-the-art alternative learning methods.

We also extensively compared our approach to a recently proposed new memory-based learning algorithm, memory-based sequence learning (MBSL, [Argamon *et al.*, 1998] and showed that MBL, which is a computationally simpler algorithm than MBSL, is able to reach similar precision and recall when restricted to the MBSL definition of the NP chunking, subject detection and object detection tasks. More importantly, MBL is more flexible in the definition of the shallow parsing tasks: it allows nested relations to be detected; it allows the addition and integration into the task of various additional sources of information apart from POS tags; it can segment a tagged sentence into different types of constituent chunks in one pass; it can scan a chunked sentence for different relation types in one pass (though separating subject-verb detection from object-verb detection is surely an option that must be investigated).

In current research we are extending the approach to other types of constituent chunks and other types of syntactic relations. Combined with previous results on PP-attachment [Zavrel *et al.*, 1997], the results presented here will be integrated into a complete shallow parser.

Acknowledgements

This research was carried out in the context of the "Induction of Linguistic Knowledge" (ILK) research programme, supported partially by the Foundation of Language, Speech and Knowledge (TSL), which is funded by the Netherlands Organisation for Scientific Research (NWO). The authors would like to thank the other members of the ILK group for the fruitful discussions and comments.

References

[Abney, 1996] S. Abney. Statistical methods and linguistics. In Judith L. Klavans and Philip Resnik, editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pages 1–26. MIT Press, Cambridge, MA, 1996.

[Argamon *et al.*, 1998] S. Argamon, I. Dagan, and Y. Krymolowski. A memory-based approach to learning shallow natural language patterns. In *Proc. of 36th annual meeting of the ACL*, pages 67–73, Montreal, 1998.

[Bod, 1995] R. Bod. Enriching linguistics with statistics: Performance models of natural language. Dissertation, ILLC, Universiteit van Amsterdam, 1995.

[Brill, 1992] E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of the Third ACL Applied NLP*, pages 152–155, Trento, Italy, 1992.

[Buchholz, 1998] Sabine Buchholz. Distinguishing complements from adjuncts using memory-based learning. In *Proceedings of the ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*, 1998.

[Cardie and Pierce, 1998] C. Cardie and D. Pierce. Error-driven pruning of treebank grammars for base noun phrase identification. In *Proc. of 36th annual meeting of the ACL*, pages 218–224, Montreal, 1998.

[Cardie, 1994] C. Cardie. *Domain Specific Knowledge Acquisition for Conceptual Sentence Analysis.* PhD thesis, University of Massachusets, Amherst, MA, 1994.

[C.J.van Rijsbergen, 1979] C.J.van Rijsbergen. *Information Retrieval.* Buttersworth, London, 1979.

[Daelemans *et al.*, 1996] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. MBT: A memory-based part of speech tagger generator. In E. Ejerhed and I. Dagan, editors, *Proc. of Fourth Workshop on Very Large Corpora*, pages 14–27. ACL SIGDAT, 1996.

[Daelemans *et al.*, 1999a] W. Daelemans, A. Van den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning*, 34:11–41, 1999.

[Daelemans *et al.*, 1999b] W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch. TiMBL: Tilburg Memory Based Learner, version 2.0, reference manual. Technical Report ILK-9901, ILK, Tilburg University, 1999.

[Daelemans, 1995] W. Daelemans. Memory-based lexical acquisition and processing. In P. Steffens, editor, *Machine Translation and the Lexicon*, Lecture Notes in Artificial Intelligence, pages 85–98. Springer-Verlag, Berlin, 1995.

[Dedina and Nusbaum, 1991] M. J. Dedina and H. C. Nusbaum. PRONOUNCE: a program for pronunciation by analogy. *Computer Speech and Language*, 5:55–64, 1991.

[Grefenstette, 1996] Gregory Grefenstette. Light parsing as finite-state filtering. In Wolfgang Wahlster, editor, *Workshop on Extended Finite State Models of Language, ECAI'96, Budapest, Hungary.* John Wiley & Sons, Ltd., 1996.

[Magerman, 1994] D. M. Magerman. Natural language parsing as statistical pattern recognition. Dissertation, Stanford University, 1994.

[Marcus *et al.*, 1993] M. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[Ramshaw and Marcus, 1995] L.A. Ramshaw and M.P. Marcus. Text chunking using transformation-based learning. In *Proc. of third workshop on very large corpora*, pages 82–94, June 1995.

[Ratnaparkhi, 1997] A. Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. Technical Report cmp-lg/9706014, Computation and Language, http://xxx.lanl.gov/list/cmp-lg/, June 1997.

[Veenstra, 1998] J. B. Veenstra. Fast np chunking using memory-based learning techniques. In *Proceedings of BENELEARN'98*, pages 71–78, Wageningen, The Netherlands, 1998.

[Vilain and Day, 1996] M.B. Vilain and D.S. Day. Finite-state phrase parsing by rule sequences. In *Proc. of COLING*, Copenhagen, 1996.

[Yvon, 1996] F. Yvon. *Prononcer par analogie: motivation, formalisation et évaluation.* PhD thesis, Ecole Nationale Supérieure des Télécommunication, Paris, 1996.

[Zavrel and Daelemans, 1997] J. Zavrel and W. Daelemans. Memory-based learning: Using similarity for smoothing. In *Proc. of 35th annual meeting of the ACL*, Madrid, 1997.

[Zavrel *et al.*, 1997] J. Zavrel, W. Daelemans, and J. Veenstra. Resolving pp attachment ambiguities with memory-based learning. In M. Ellison, editor, *Proc. of the Workshop on Computational Language Learning (CoNLL'97)*, ACL, Madrid, 1997.