# Finding Representations for Memory-Based Language Learning

**Stephan Raaijmakers**
raaijmakers@inl.nl
Institute for Dutch Lexicology (INL)
P.O. Box 9515
2300 RA Leiden
The Netherlands

## Abstract

Constructive induction transforms the representation of instances in order to produce a more accurate model of the concept to be learned. For this purpose, a variety of operators has been proposed in the literature, including a Cartesian product operator forming pairwise higher-order attributes. We study the effect of the Cartesian product operator on memory-based language learning, and demonstrate its effect on generalization accuracy and data compression for a number of linguistic classification tasks, using $k$-nearest neighbor learning algorithms. These results are compared to a baseline approach of backward sequential elimination of attributes. It is demonstrated that neither approach consistently outperforms the other, and that attribute elimination can be used to derive compact representations for memory-based language learning without noticeable loss of generalization accuracy.

## Introduction

It is a widely held proposition that inductive learning models, such as decision trees [Quinlan, 1993] or $k$-nearest neighbor models [Aha, Kibler & Albert, 1991], are heavily dependent upon their representational biases. Both decision tree algorithms and instance-based algorithms have been reported to be vulnerable to irrelevant or noisy attributes in the representation of exemplars, which unnecessarily enlarge the search space for classification [John, 1997]. In general, there are two options for dealing with this problem. Attribute elimination (or selection) can be applied in order to find a minimal set of attributes that is maximally informative for the concept to be learned. Attribute elimination can be seen as a radical case of attribute weighting [Scherf & Brauer, 1997, Aha, 1998], where attributes are weighted on a binary scale, as either relevant or not; more fine-grained methods of attribute weighting take information-theoretic notions into account such as *information gain ratio* [Quinlan, 1993. Daelemans *et al.*, 1997]. Successful attribute elimination leads to compact datasets, which possibly increase classification speed. Constructive induction, on the other hand, tries to exploit dependencies between attributes, by combining them into complex attributes that increase accuracy of the classifier. For instance-based algorithms, this approach has been demonstrated to correct invalid independence assumptions made by the algorithm [Pazzani, 1998]: e.g., for the Naive Bayes classifier (Duda & Hart, 1973), the unwarranted assumption that in general the various attributes $a_i = v_i$ are independent, and form a joint probability model for the prediction of the class $C$:

$$P(C \mid a_1 = v_1 \wedge \ldots \wedge a_n = v_n) = \frac{P(C)\prod P(a_i = v_i \mid C)}{P(a_1 = v_1 \wedge \ldots \wedge a_n = v_n)} \quad (1)$$

Constructive induction thus can be used to invent relationships between attributes that, apart from possibly offering insight into the underlying structure of the learning task, may boost performance of the resulting classifier. Linguistic tasks are sequential by nature, as language processing is a linear process, operating on sequences with a temporal structure (see e.g. Cleeremans (1993) for motivation for the temporal structure of finite-state grammar learning). Learning algorithms like $k$-nearest neighbor or decision trees abstract away from this linearity, by treating representations as multisets of attribute-value pairs, i.e. permutation-invariant lists. Using these algorithms, constructive induction cannot be used for corrections on the linearity of the learning task, but it can be used to study attribute interaction irrespective of ordering issues.

In this paper, the use of constructive induction is contrasted with attribute elimination for a set of linguistic learning tasks. The linguistic learning domain appears to be deviant from other symbolic domains in being highly susceptible to editing. It has been noticed [Daelemans *et al.*, 1999i] that editing exceptional

instances from linguistic instance bases tends to harm generalization accuracy. In this study, we apply editing on the level of instance representation. The central question is whether it is more preferable to correct linguistic tasks by combining (possibly noisy or irrelevant) attributes, or by finding informative subsets.

## Representation Transformations

John (1997) contains presentations of various attribute selection approaches. In Yang & Honovar (1998), a genetic algorithm is used for finding informative attribute subsets, in a neural network setting. Cardie (1996) presents an attribute selection approach to natural language processing (relative pronoun disambiguation) incorporating a small set of linguistic biases (to be determined by experts).

Many operators have been proposed in the literature for forming new attributes from existing ones. Pagallo & Hauser (1990) propose boolean operators (like conjunction and negation) for forming new attributes in a decision tree setting. Aha (1991) describes IB3-CI, a constructive indiction algorithm for the instance-based classifier IB3. Aiming at reducing similarity between an exemplar and its misclassifying nearest neighbor, IB3-CI uses a conjunctive operator forming an attribute that discriminates between these two. Bloedorn & Michalski (1991) present a wide variety of mathematical and logical operators within the context of the AQ17-DC1 system. A general perspective on constructive induction is sketched in Bloedorn, Michalski & Wnek (1994). Keogh & Pazzani (1999) propose correlation arcs between attributes, augmenting Naive Bayes with a graph structure.

Pazzani (1998) proposes a Cartesian product operator for joining attributes, and compares its effects on generalization accuracy with those of attribute elimination, for (a.o.) the Naive Bayes and PEBLS (Cost & Salzberg, 1993) classifiers. The Cartesian product operator joins two attributes $A_1$ and $A_2$ into a new, complex attribute $A_1 \cdot A_2$, taking values in the Cartesian product

$$\{< a_i, a_j > | \; a_i \in Values(A_1) \wedge a_j \in Values(A_2)\} \quad (2)$$

where $Values(A)$ is the value set of attribute $A$. The Cartesian product operator has an intrinsic linear interpretation: two features joined in a Cartesian product form an ordered pair with a precedence relation (the ordered pair $< a, b >$ differs from the ordered pair $< b, a >$). This linear interpretation vanishes in learning algorithms that do not discern internal structure in attribute values (like standard nearest neighbor).

Pazzani's *backward sequential elimination and joining* algorithm (BSEJ) finds the optimal representation transformation by considering each pair of attributes in turn, using leave-one-out cross-validation to determine the effect on generalization accuracy. Attribute joining carries out an implicit but inevitable elimination step: wiping out an attribute being subsumed by a combination. This reduces the dimensionality of the result dataset with one dimension. Following successful joining, the BSEJ algorithm carries out an *explicit* elimination step, attempting to delete every attribute in turn (including the newly constructed attribute) looking for the optimal candidate using cross-validation. The algorithm converges when no more transformations can be found that increase generalization accuracy. This approach is reported to produce significant accuracy gain for Naive Bayes and for PEBLS. Pazzani contrasts BSEJ with a *backward sequential elimination* algorithm (BSE, *backward sequential elimination*, progressively eliminating attributes (and thus reducing dimensionality) until accuracy degrades. He also investigates forward variants of these algorithms, which successively build more complex representations up to convergence. Both for PEBLS and Naive Bayes, attribute joining appears to be superior to elimination, and the backward algorithms perform better than the forward algorithms. For $k$-nearest neighbor algorithms based on the unweighted overlap metric, BSEJ did not outperform BSE.

Conditioning representation transformations on the performance of the original classifier implements a wrapper approach (John, 1997; Kohavi & John, 1998), which has proven an accurate, powerful method to measure the effects of data transformations on generalization accuracy. The transformation process is wrapped around the classifier, and no transformation is carried out that degrades generalization accuracy.

In this study, two algorithms, an implementation of BSE and a simplification of the BSEJ algorithm, were wrapped around three types of classifiers: IB1-IG, IB1-IG&MVDM (a classifier related to PEBLS in using MVDM) and IGTREE [Daelemans *et al.*, 1997]. All of these classifiers are implemented in the TiMBL package [Daelemans *et al*, 1999ii]. IB1-IG is a $k$-nearest neighbor algorithm using a weighted overlap metric, where the attributes of instances have their information gain ratio as weight. For instances $X$ and $Y$, distance is computed as

$$\Delta(X, Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i) \quad (3)$$

where $\delta$ is the overlap metric, and $w_i$ is the information gain ratio (Quinlan, 1993) of attribute $i$.

The PEBLS algorithm can be approximated to a certain extent by combining IB1-IG with the Modified Value Difference Metric (MVDM) of Cost & Salzberg

(1993). The MVDM defines the difference between two values $x$ and $y$ respective to a class $C_i$ as

$$\delta(x,y) = \sum_{i=1}^{n} \mid P(C_i \mid x) - P(C_i \mid y) \mid \qquad (4)$$

i.e., it uses the probabilities of the various classes conditioned on the two values to determine overlap. Attribute weighting of IB1-IG&MVDM (information gain ratio based) differs from PEBLS: PEBLS uses performance-based weighting based on class predicyion strength, where exemplars are weighted according to an accuracy or reliability ratio.

IGTREE is a tree-based $k$-nearest neighbor algorithm, where information gain is used as a heuristic to insert nodes in the tree. For every non-terminal node, a default classification is stored for the path leading to it. Whenever no exact match can be found for an unknown instance to be classified, the default classification associated with the last matching attribute is returned as classification for the instance. Although IGTREE sometimes lags behind IB1-IG in accuracy, it provides for much faster, high quality classifiers.

An implementation of the BSE algorithm is outlined in figure . It is akin in spirit to the backward elimination algorithm of John (1997). During every pass, it measures the effects on generalization accuracy of eliminating every attribute in turn, only carrying out the one which maximizes accuracy. A simplified version of the BSEJ algorithm called *backward sequential joining with information gain ratio* (BSJ-IG) is outlined in figure . It checks the $\frac{N!}{2(N-2)!}$ ordered combinations for $N$ features during each pass, and carries out the one resulting in the maximum gain in accuracy (as a consequence of the permutation invariance, the total search space of $\frac{N!}{(N-2)!}$ possible combinations can be halved). Any two joined attributes are put on the position with the maximum information gain ratio of both original positions, after which the remaining candidate position is wiped out. Again, as the used classifiers are all permutation-invariant with respect to their representations, this is only a decision procedure to find a target position for the attribute combination; all candidate positions are equivalent target positions.

Unlike the original BSEJ algorithm, BSJ-IG omits the additional explicit attribute elimination step directly after every attribute joining step, in order to segregate the effects of attribute joining as much as possible from those of attribute elimination.

Both BSE and BSJ-IG algorithms are hill-climbing algorithms. and, as such, are vulnerable to local minima. Ties are resolved randomly by both.

## Experiments

The effects of forming Cartesian product attributes on generalization accuracy and reduction of dimensionality (compression) were compared with those of backward sequential elimination of attributes. The following 7 linguistic datasets were used. STRESS is a selection of secondary stress assignment patterns from the Dutch version of the Celex lexical database [Baayen, Piepenbrock & van Rijn, 1993], on the basis of phonemic representations of syllabified words. Attribute values are phonemes. Also derived from Celex is the DIMIN task, a selection of diminutive formation patterns for Dutch. This task consists of assigning Dutch diminutive suffixes to a noun, based on phonetic properties of (maximally) the last three syllables of the noun. Attribute values are phoneme representations as well as stress markers for the syllables. The WSJ-NPVP set consists of part-of speech tagged Wall Street Journal material (Marcus, Santorini & Marcinkiewicz, 1993), supplemented with syntactic tags indicating noun phrase and verb phrase boundaries (Daelemans *et al* , 1999iii). WSJ-POS is a fragment of the Wall Street Journal part-of-speech tagged material (Marcus, Santorini and Marcinkiewicz, 1993). Attributes values are parts of speech, which are assigned using a windowing approach, with a window size of 5. INL-POS is a part-of-speech tagging task for Dutch, using the Dutch-Tale tagset [van der Voort van der Kleij *et al.*, 1994], attribute values are parts of speech. Using a windowing approach, on the basis of a 7-cell window, part of speech tags are disambiguated. GRAPHON constitutes a grapheme-to-phoneme learning task for English, based on the Celex lexical database. Attribute values are graphemes (single characters). to be classified as phonemes. PP-ATTACH, finally, is a prepositional phrase (PP) attachment task for English, where PP's are attached to either noun or verb projections, based on lexical context. Attribute values are word forms for verb, the head noun of the following noun phrase, the preposition of the following PP, and the head noun of the PP-internal noun phrase (like `bring attention to problem`). The material has been extracted by Ratnaparkhi *et al.* (1994) from the Penn Treebank Wall Street Journal corpus. Key numerical characteristics of the datasets are summarized in table 1.

Each of these datasets was subjected to the BSJ-IG and the BSE wrapper algorithms, embedding either the IB1-IG or IGTREE architecture. Both the Naive Bayes and PEBLS classifier investigated by Pazzani (1998) allow for certain frequency tendencies hidden in the data to bear on the classification. This has a smoothing effect on the handling of low-frequency events. which benefit from analogies with more reliable higher-frequency

```
Procedure BSE
Input:  a training set T
Output: a new training set T' with possibly attributes removed

Set Acc to Accuracy(T) for the current classifier
Set Success to true
While (Success) do
      Set Success to false
      For every attribute A in T do
          Produce T' by removing A from every instance in T
          NewAcc=Accuracy(T') for the current classifier
          If (NewAcc≥Acc)
          Then
             Set Acc to NewAcc
             Set Winner to T'
             Set Success to true
      If Success equals true
      Then
          Set T to Winner
Return T
```

Figure 1: A wrapper implementation of Backward Sequential Elimination (BSE).

events. In order to assess the effects of smoothing, the following additional experiments were carried out. Embeddded into BSE and BSJ-IG, the PEBLS approximation IB1-IG with MVDM was applied to three datasets: STRESS, DIMIN and PP-ATTACH, for three values of $k$ (1, 3, 7), the size of the nearest neighbor set. Values for $k$ larger than 1, i.e. non-singleton nearest neighbor sets. have been found to reproduce some of the smoothing inherent to statistical back-off models (Daelemans *et al.*. 1999ii; Zavrel & Daelemans, 1997).

Generalization accuracy for every attribute joining or elimination step was measured using 10-fold cross-validation, and significance was measured using a two-tailed paired t-test at the .05 level. All experiments were carried out on a Digital Alpha XL-266 (Linux) and a Sun UltraSPARC-IIi (Solaris). Due to slow performance of the IB1-IG model on certain datasets with the used equipment, IB1-IG experiments with WSJ-NPVP could not be completed.

## Results

The results show, first of all, that the compression rates obtained with BSE (average 34.9%) were consistently higher than those obtained with BSJ-IG (average 28.6%) (table 2).

Secondly, BSE and BSJ-IG have comparable effects on accuracy. BSE generally boosts IGTREE performance to IB1-IG level, and leads to significant accuracy gains for two datasets, STRESS and PP-ATTACH (tabel 3). BSJ-IG does so for the STRESS set (tabel 4). Neither BSE nor BSJ-IG produce any significant gain in accuracy for the IB1-IG classifier. This generalizes the findings of Pazzani (1998) for classifiers based on unweighted overlap metrics to classifiers based on a weighted overlap metric.

For the classifier IB1-IG&MVDM. the situation is more complex (table 5). First, for $k = 1$. BSE and BSJ-IG have comparable accuracy. For the STRESS and PP-ATTACH sets, both algorithms produce significant and comparable accuracy gains. Second, compression by BSE is significantly higher than compression by BSJ-IG (47.2% vs. 30.6%).

For the larger values for $k$ (3, 7), BSJ-IG produces significant higher accuracies on the STRESS set, outperforming BSE. Moreover, BSJ-IG yields a compression rate comparable to BSE. BSE compression drops from 47.2% to 27.8%.

A detailed look at the representations produced by BSE and BSJ-IG reveals the following.

```
Procedure BSJ-IG
Input:  a training set T
Output: a new training set T' with possibly higher-order induced attributes

Set Acc to Accuracy(T) for the current classifier
Set Success to true
While (Success) do
      Set Success to false
      For every ordered combination of two attributes A_i and A_j in T do
          Produce T' from T by joining A_i and A_j, putting them on the position
                  k ∈ {i, j} with the largest information gain ratio.
          NewAcc=Accuracy(T') for the current classifier
          If (NewAcc≥Acc)
          Then
             Set Acc to NewAcc
             Set Winner to T'
             Set Success to true
      If Success equals true
      Then
          Set T to Winner
Return T
```

Figure 2: A wrapper implementation of Backward Sequential Joining with Information Gain ratio (BSJ-IG)

- (BSJ-IG) IB1-IG&BSJ-IG and IGTREE&BSJ-IG only agree on WSJ-POS: they both join the same attributes. For the other datasets, there is no overlap at all.
- (BSE) For the WSJ-POS set, BSE deletes exactly the same two features that are joined by BSJ-IG for IB1-IG and IGTREE. For the DIMIN set, IB1-IG&BSE and IGTREE&BSE delete 4 common features. For STRESS, all features deleted by IB1-IG&BSE are deleted by IGTREE&BSE as well. On the INL-POS set, three common features are deleted. Frequently, BSE was found to delete an attribute joined by BSJ-IG.
- (IB1-IG&MVDM, BSJ-IG) BSJ-IG produces no overlap for DIMIN for the three different classifiers ($k = 1, 3, 7$). For STRESS, the $k = 1$, $k = 3$ and $k = 7$ classifiers join one common pair of attributes. This is the pair consisting of the nucleus and coda of the last syllable, indeed a strong feature for stress assignment (Daelemans, p.c.). For PP-ATTACH, the $k = 1$, $k = 3$ and $k = 7$ classifiers identify attribute 4 (the head noun of the PP-internal noun phrase) for joining with another attribute. Attribute 4 clearly introduces sparseness in the dataset: it has 5,695 possible values, opposed to maximally 4,405 values for the other attributes. The $k = 3$ and $k = 7$ classifiers agree fully here.
- (IB1-IG&MVDM, BSE) On the DIMIN set, the $k = 1$ and $k = 3$ classifiers differ in 1 attribute elimination only. They display no overlap with $k = 7$, which eliminates entirely other attributes. For STRESS, $k = 1$ and $k = 3$ classifiers overlap on 3 attributes. The three classifiers delete 1 common attribute (not the nucleus or coda). For PP, the $k = 3$ and $k = 7$ classifiers do not eliminate attributes; the $k = 1$ classifier deletes the attribute 4 (PP-internal head noun), and even the first verb-valued attribute. In doing so, it constitutes a strongly lexicalised model for PP-attachment taking only into account the first head noun and the following preposition.

BSE produced more overlapping results across classifiers than BSJ-IG. IB1-IG&MVDM with BSJ-IG is the only type of classifier that is able to trap the important interaction between nucleus and coda in the STRESS set. Due to lack of domain knowledge, we cannot be certain that other important interactions have not been

| Dataset | Instances | Attributes | IB1-IG | IGTREE |
|---|---|---|---|---|
| STRESS | 3,000 | 12 | 85.9±0.8 | 81.6±1.0 |
| DIMIN | 3,000 | 12 | 98.2±0.4 | 98.2±0.5 |
| WSJ-NPVP | 200,000 | 8 | 97.1±0.08 | 96.5±0.08 |
| GRAPHON | 350,000 | 7 | 96.6±0.04 | 96.2±0.06 |
| WSJ-POS | 399,925 | 5 | 95.9±0.04 | 95.9±0.04 |
| INL-POS | 250,004 | 7 | 96.3±0.1 | 96.3±0.1 |
| PP-ATTACH | 20,801 | 4 | 81.3±0.5 | 78.3±0.4 |

Table 1: Number of instances, attributes and original accuracies for the datasets.

| Algorithm | BSE | BSJ-IG |
|---|---|---|
| IB1-IG | 34.2 | 23 |
| IGTREE | 34.7 | 30.9 |
| IB1-IG&MVDM, k=1 | 47.2 | 30.6 |
| IB1-IG&MVDM, k=3 | 30.5 | 33.3 |
| IB1-IG&MVDM, k=7 | 27.8 | 25 |
| Average | 34.9 | 28.6 |

Table 2: Average compression rates.

trapped as well; this lies outside the scope of this study. Although firm conclusions cannot be drawn on the basis of three datasets only, the compact and accurate results of the $k = 3$ and $k = 7$ classifiers may indicate a tendency for smoothing algorithms to compensate better for eventual non-optimal attribute combinations than for eliminated attributes. This would be in agreement with Pazzani's findings for PEBLS and Naive Bayes.

Frequently, cases were observed where BSE eliminates attributes that were used for joining by BSJ-IG. This indicates that at least some of the advantages of attribute joining originate from implicit attribute elimination rather than combination, which has also been noted by Pazzani (1998): removing an attribute may improve accuracy more than joining it to another attribute.

## Conclusions

The effects of two representation-changing algorithms on generalization accuracy and data compression were tested for three different types of nearest neighbor classifiers, on 7 linguistic learning tasks. As a consequence of the permutation-invariance of the used classifiers and the use of hill-climbing algorithms, a practical sampling of the search space of data transformations was applied. BSE, an attribute elimination algorithm, was found to produce accurate classifiers, with consistently higher data compression rates than BSJ-IG, an attribute joining algorithm. The generalization accuracy of BSE is comparable to that of BSJ-IG.

Some evidence hints that attribute joining may be more succesful – both for compression and accuracy – for classifiers employing smoothing techniques, e.g. PEBLS-like algorithms which select a nearest neighbor from a nearest neighbor set using frequency information. This type of classifier was able to trap at least one important attribute interaction in the STRESS domain, offering extended insight in the underlying learning task. Further evidence is needed to confirm this conjecture, and may shed further light on the question whether and how linguistic learning tasks could benefit from attribute interaction. An alternative line of research to be pursued will address classifier models that allow for linear encoding of linguistic learning tasks: these models will allow investigations into corrections on the linearity of linguistic tasks.

## Acknowledgements

I would like to thank Michael Pazzani for helpful comments, as well as the anonymous reviewers. Thanks go to the Induction of Linguistic Knowledge group of Tilburg University (Antal van den Bosch, Sabine Buchholz, Walter Daelemans, Jorn Veenstra and Jakub Zavrel) for valuable feedback and datasets. INL is acknowledged for the INL-POS dataset and access to Sun equipment.

## References

[Aha, 1991] Aha, D. (1991). Incremental constructive induction: an instance-based approach. *Proceed-*

| Dataset | IB1-IG&BSE | | IGTREE&BSE | |
|---|---|---|---|---|
| STRESS | 10 | 86.2±1.0 | 6 | 84.9±1.0+ |
| DIMIN | 4 | 98.5±0.4 | 5 | 98.4±0.5 |
| WSJ-NPVP | - | - | 7 | 96.9±0.07 |
| GRAPHON | = | = | = | = |
| WSJ-POS | 3 | 96.1±0.04 | 3 | 96.1±0.04 |
| INL-POS | 3 | 96.5±0.09 | 3 | 96.5±0.09 |
| PP-ATTACH | 3 | 82.0±0.7 | 3 | 81.9±0.7+ |

Table 3: Number of remaining attributes and accuracies for BSE. A '+' indicates a significant increase in accuracy compared to the original algorithm; a '-' indicates the experiment could not be completed.

| Dataset | IB1-IG&BSJ-IG | | IGTREE&BSJ-IG | |
|---|---|---|---|---|
| STRESS | 9 | 86.6±1.0 | 8 | 85.2±0.8+ |
| DIMIN | 6 | 98.5±0.4 | 6 | 98.4±0.4 |
| WSJ-NPVP | - | - | 6 | 96.9±0.08 |
| GRAPHON | = | = | 6 | 96.2±0.05 |
| WSJ-POS | 4 | 96.0±0.04 | 4 | 96.0±0.04 |
| INL-POS | 4 | 96.5±0.1 | 4 | 96.5±0.1 |
| PP-ATTACH | = | = | = | = |

Table 4: Number of remaining attributes and accuracies for BSJ-IG. A '+' indicates a significant increase in accuracy compared to the original algorithm; a '=' indicates no difference with respect to the original algorithm; a '-' indicates the experiment could not be completed.

*ings of the 8th Machine Learning Workshop.*

[Aha, Kibler & Albert, 1991] Aha, D. , Kibler, D. & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning, 6*, 37-66.

[Aha, 1998] Aha, D. (1998). Feature weighting for lazy learning algorithms. In Liu, H. & Motoda. H. (eds.), *Feature Extraction, Construction and Selection.* Boston: Kluwer.

[Baayen, Piepenbrock & van Rijn, 1993] Baayen, H. , Piepenbrock, R. & van Rijn, H. (1993). *The CELEX lexical database on CD-ROM.* Linguistic Data Consortium. Philadelphia, PA.

[Bloedorn&Michalski, 1991] Bloedorn, E. & Michalski, R.S. (1991). *Constructive Induction from Data in AQ17-DCI.* MLI91-12. Artificial Intelligence Center, George Mason University.

[Bloedorn, Michalski&Wnek] Bloedorn, E., Michalski, R. & Wnek, J. (1994). *Matching Methods with Problems: A Comparative Analysis of Constructive Induction Approaches.* MLI94-12. Artificial Intelligence Center, George Mason University.

[Cardie, 1996] Cardie, C. (1996). Automating feature set selection for case-based learning of linguistic knowledge. *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* University of Pennsylvania.

[Cleeremans, 1993] Cleeremans, A. (1993). *Mechanisms of implicit learning: connectionist models of sequence processing.* Cambridge, Mass.: MIT Press.

[Cost&Salzberg, 1993] Cost, S. & Salzberg, S. (1993) A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning, 10*, 57-78.

[Daelemans *et al.*, 1997] Daelemans, W., van den Bosch, A. & Zavrel, J. (1997). IGTREE: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review, 11*, 407-423.

[Daelemans *et al.*, 1999i] Daelemans, W., van den Bosch, A. & Zavrel, J. (1999i). Forgetting exceptions is harmful in language learning. *Machine Learning, 11*, 11-43.

[Daelemans *et al*, 1999ii] Daelemans. W., Zavrel, J., van der Sloot, K. & van den Bosch. A. (1999ii).

IB1-IG&MVDM

| Dataset | k=1 | k=3 | k=7 |
|---|---|---|---|
| STRESS | 86.1±1.0 | 86.8±1.0 | 87.0±1.0 |
| DIMIN | 98.0±0.4 | 98.4±0.3 | 98.4±0.3 |
| PP-ATTACH | 75.7±0.7 | 76.9±0.7 | 77.7±0.6 |

IB1-IG&MVDM&BSE

| Dataset | k=1 | | k=3 | | k=7 | |
|---|---|---|---|---|---|---|
| STRESS | 7 | 88.3±1.0+ | 7 | 88.5 ±1.0 | 9 | 88.0±1.0 |
| DIMIN | 6 | 98.6±0.4 | 6 | 98.8±0.4 | 5 | 98.7±0.2 |
| PP-ATTACH | 2 | 78.1±0.6+ | = | = | = | = |

IB1-IG&MVDM&BSJ-IG

| Dataset | k=1 | | k=3 | | k=7 | |
|---|---|---|---|---|---|---|
| STRESS | 8 | 89.0±1.0+ | 8 | 89.8±1.0+ | 10 | 89.8±0.7+ |
| DIMIN | 8 | 98.4±0.4 | 7 | 98.6±0.3 | 8 | 98.6±0.2 |
| PP-ATTACH | 3 | 77.7±0.5+ | 3 | 77.5±0.5 | 3 | 77.7±0.5 |

Table 5: Number of remaining attributes and accuracies for IB1-IG with MVDM, IB1-IG with MVDM and BSE, and for IB1-IG with MVDM and BSJ-IG, for $k$=1, 3, and 7. A '+' indicates a significant increase in accuracy compared to the original algorithm; a '=' indicates no difference with respect to the original algorithm.


*TiMBL: Tilburg Memory Based Learner, version 2.0, Reference Guide.* Tilburg: Induction of Linguistic Knowledge. Available from `http://ilk.kub.nl/~ilk/papers/ilk9901.ps.gz`.

[Daelemans, Buchholz & Veenstra, 1999iii] Daelemans, W., Buchholz, S. & Veenstra, J. (1999iii). Memory-based shallow parsing. Paper ILK, Tilburg.

[Duda&Hart, 1973] Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis.* Wiley Press.

[John, 1997] John, G. H., 1997. *Enhancements to the data mining process.* Ph.D. dissertation. Stanford University.

[Kohavi & John, 1998] Kohavi, R. & John, G.H. (1998). The wrapper approach. In Liu, H. & Motoda, H. (eds.), *Feature Extraction, Construction and Selection.* Boston: Kluwer.

[Keogh&Pazzani, 1999] Keogh, E. J. & M. J. Pazzani, 1999. Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches. *Proceedings 7th International Workshop on AI and Statistics*, 225-230. Ft. Lauderdale, Florida.

[Marcus, Santorini & Marcinkiewicz, 1993] Marcus, M., Santorini, B. & Marcinkiewicz (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics 19*: 313-330.

[Pagallo & Hauser, 1990] Pagallo, G., Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine Learning, 5*, 71-99.

[Pazzani, 1998] Pazzani, Michael J. (1998). Constructive Induction of Cartesian Product Attributes. In Liu, H. & Motoda, H. (eds.), *Feature Extraction, Construction and Selection.* Boston: Kluwer.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA.: Morgan Kaufmann.

[Ratnaparkhi *et al.*, 1994] Ratnaparkhi, A., Reynar, J. & Roulos, S. (1994). A maximum entropy model for Prepositional Phrase Attachment. *Proceedings ARPA Workshop on Human Language Technology.* Plainsboro.

[Scherf & Brauer, 1997] Scherf, M., Brauer, W. (1997). Feature selection by means of a feature weighting approach. Technical report No. FKI-221-97, Forschungsberichte Künstliche Intelligenz. Institut für Informatik, Technische Universität München.

[van der Voort van der Kleij *et al.*, 1994] van der Voort van der Kleij, J., Raaijmakers, S., Panhuijsen, M., Meijering, M., Sterkenburg, R.

van (1994). Een automatisch geanalyseerd corpus hedendaags Nederlands in een flexibel retrievalsysteem. (*An automatically analysed corpus of contemporary Dutch in a flexible retrieval system*, in Dutch.) *Proceedings Informatiewetenschap 1994*. Tilburg.

[Yang & Honovar, 1998] Yang, J. & Honovar, V. (1998). Feature subset selection using a genetic algorithm. In Liu, H. & Motoda, H. (eds.), *Feature Extraction, Construction and Selection*. Boston: Kluwer.

[Zavrel & Daelemans, 1997] Zavrel, J. & Daelemans, W. (1997). Memory-based learning: using similarity for smoothing. *Proceedings of the 35th annual meeting of the ACL*. Madrid.