Toward an Exemplar-Based Computational Model for Cognitive Grammar

Walter Daelemans*

Abstract

An exemplar-based computational framework is presented which is compatible with Cognitive Grammar. In an exemplar-based approach, language acquisition is modeled as the incremental, data-oriented storage of experiential patterns, and language performance as the extrapolation of information from those stored patterns on the basis of a language-independent information-theoretic similarity metric. We show that this simple architecture works for many aspects of phonological, morphological, and morphosyntactic acquisition and processing. Furthermore, we sketch how the approach may also work for syntactic processing. A central insight of the approach, based on the results of computational modeling experiments, is that abstraction of representations is not only unnecessary to achieve generalization (i.e. to make the system productive, and to make it go 'beyond' the learned patterns), but even harmful, and that useful language-independent metrics can be found for defining similarity in the context of language processing.

In: van der Auwera et al. (eds.) English as a Human Language
Muenchen: LINCOM, 73-82, 1998

1 Usage-based Models of Language Structure

In the generative tradition, generality is achieved by means of abstraction, and the representations of choice to describe these abstractions are rules. This implies that redundancy and the storage of individual instances are to be avoided, except for exceptions to the generalizations expressed in rules. In Langacker, 1991 (Chapter 10), this methodology is critically examined, and cognitive grammar is described as an alternative usage-based model of language structure. In the latter, bottom-up, approach, patterns (rules, generalizations) and (redundant) instantiations of those rules are assumed to co-exist in the grammar, describing phenomena at all levels of generality, from exceptionless regularities to idiosyncratic exceptions. Rules are presumed to be necessary for the computation of novel instantiations.

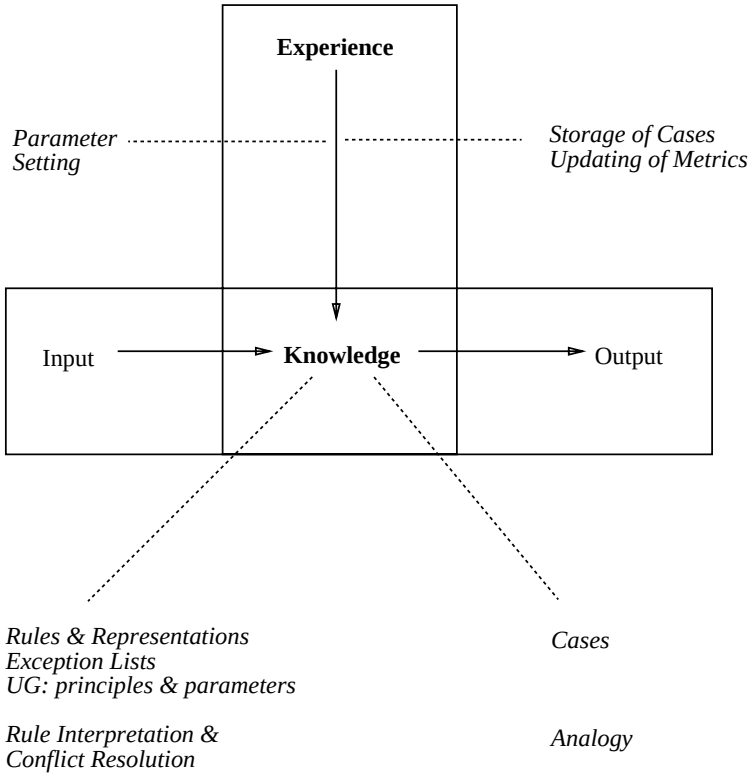
In the remainder of this paper we will introduce an exemplar-based approach to language acquisition and processing. The approach is in large part compatible with Langacker's usage-based model, but is more radical in its "maximalism": language knowledge is supposed to consist *only* of "instantiations" (exemplars); there is no role for explicit abstractions corresponding to (sub)regularities. We will argue on the basis of computational modeling experiments that the adoption of abstractions (rules, patterns), taken as necessary for explaining generalization and productivity in both the generative and the cognitive grammar approach, is misguided. Furthermore, the exemplar-based approach contributes to making cognitive grammar ideas more concrete by providing computational operationalisations of both acquisition and processing in such a framework.

*Affiliations: Tilburg University, ILK group, Computational Linguistics, and University of Antwerp, CNTS.

2 An Exemplar-Based Linguistic Theory

The Chomskian view on the acquisition and nature of linguistic knowledge is familiar: every individual is born with a 'universal grammar' as part of her genetic endowment, consisting of rules (principles), and parameters with a predefined default value and a number of possible settings. In language acquisition, the parameters are set to their correct values for the grammar of the specific language being learned, based on cues in the language experience the individual is exposed to. A theoretical description of the language knowledge in this framework takes the form of abstractions (rules or principles) manipulating symbolic representations, and exception lists for irregular phenomena. When we want to use such an architecture for language processing (i.e. when we want to give a psycholinguistic performance interpretation to this architecture), procedures have to be defined based on rule interpretation and conflict resolution for language generation (from logical form to phonetic form) and language comprehension (from phonetic form to logical form). Conflict resolution (deciding which rule or exception is applicable) usually takes the form of rule ordering or some form of the elsewhere condition (i.e. some form of non-monotonic reasoning).

In contrast to this nativist, rule-based approach, the exemplar-based theory proposed here is empiricist in its acquisition method and based on memory traces (patterns) rather than on explicit symbolic rules in its knowledge representation. We will call these patterns exemplars. The main differences between both architectures are summarized in Figure 1, where the vertical arrow denotes acquisition, and the horizontal arrows performance. The left part of the figure describes the traditional view (parameter setting for language acquisition; rules and exception lists for language knowledge representation; rule interpretation and conflict resolution for language processing), the right part of the figure the exemplar-based alternative, with storage of exemplars and computation of language-independent metrics for acquisition, stored exemplars for representation, and analogy for processing.



In an exemplar-based theoretical framework, *Language Acquisition* is reduced to the incremental storage of exemplars of performance-level tasks. E.g. in learning the past tense of verbs, to take a typical example, patterns of verb stems and their associated past tense forms are incrementally stored in memory as they present themselves in the language experience of the learner. How such associated pairs can be singled out from language experience can be explained in this case by referring to general principles of similarity and association: related verb forms are both formally and semantically similar. We will not go into the computational modeling of this process, however, and start in our experiments from associated pairs of 'input' and 'output'. An experience-driven theory is therefore of necessity performance-oriented (task-oriented): what is stored are exemplars of input-output associations, not abstract, reusable, task-independent generalizations.

Generalization and abstraction are considered often as indistinguishable concepts in cognitive science and linguistics: rule-based, stochastic, and connectionist approaches all work from the presupposition that they are able to generalize to new exemplars *because* of the fact that they abstract away from the data. This presupposition is false: the experiences themselves, combined with an analogical reasoning mechanism are able to generalize (handle new, previously unseen exemplars) equally well, even better as we will show in the experimental results section.

In *Language Performance*, each (sub)task is represented as a set of exemplars (experiences, cases) in memory, which act as models to new input. Each exemplar consists of an input representation and an output representation. Inputs are vectors of symbolic features describing the input representation (in our past tense example, e.g., the segmental and syllable structure information of the stem). Outputs can be boundary types (e.g. in a segmentation task), or symbolic classes (e.g. in a disambiguation task). In case of our example, outputs could be past tense suffixes or forms. New instances of the task are solved either through memory retrieval or by similarity-based (analogical) reasoning. In our example of predicting the past tense form of a verb stem, if the stem is present in memory for this task, the associated output is retrieved (or the most probable output). If it has not been encountered yet, the best matching exemplars in memory are used to extrapolate from. The computation of similarity is therefore essential for the performance of such an architecture. It is our hypothesis that language-independent similarity metrics can and should be used to compute the best matching exemplars. We will return to this in the next section.

As far as knowledge representation is concerned, there is no representational difference in an exemplar-based model between regularities, subregularities, and exceptions. As there are only exemplars in memory, and no explicit rules, there is no need for rule ordering or some form of non-monotonic reasoning. In case of ambiguity, when several different outputs are associated with the same input, the most probable solution is chosen (or a probabilistic random choice is made). Rule-like behaviour is therefore a side-effect of the interaction between the analogical reasoning process and the contents of memory. In other words, the contents of memory can be approximated as a set of rules for convenience, but these rules have no ontological status in the model (and shouldn't have in a linguistic theory).

3 Computational Modeling

Storing memory traces in the form of exemplars, and combining them with analogical reasoning in computational models of problem solving is by no means a new idea. In Artificial Intelligence, the concept has appeared in several disciplines (from computer vision to robotics), using terminology such as similarity-based, example-based, memory-based, case-based, analogical, lazy, nearest-neighbour, and instance-based (Stanfill and Waltz, 1986; Kolodner, 1993; Aha et al. 1991; Salzberg, 1990). Ideas about this type of

analogical reasoning are rare in linguistics and pyscholinguistics (Skousen, 1989; Derwing & Skousen, 1989; Chandler, 1992; Scha, 1992 are salient examples). In computational linguistics (apart from incidental computational work of the linguists referred to earlier), the general approach has recently gained some popularity: e.g., Cardie (1994, syntactic and semantic disambiguation); Daelemans (1995, an overview of work in the early nineties on memory-based computational phonology and morphology); Jones (1996, an overview of example-based machine translation research); Federici and Pirrelli (1996).

3.1 Similarity Metric

Accuracy of an exemplar-based system on previously unseen inputs crucially depends on the similarity metric (or distance metric) used. The most straightforward distance metric would be the one in Equation (1), where X and Y are the patterns to be compared (input parts of exemplars), and $\delta(x_i, y_i)$ is the distance between the values of the i-th feature in a pattern with n features.

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

Distance between two values is measured using Equation 2, an overlap metric for symbolic features (we have no numeric features in our linguistic data sets).

$$\delta(x_i, y_i) = 0 \; if \; x_i = y_i, \; else \; 1 \tag{2}$$

We will refer to this approach as IB1 (Aha et al., 1991). We extended the algorithm described there in the following way: in case an exemplar is associated with more than one output in its memory (i.e. the exemplar is ambiguous), the distribution of patterns over the different output categories is kept, and the most frequently occurring category is selected when the ambiguous exemplar is used to extrapolate from.

3.2 Feature Relevance Weighting

In this distance metric, all features describing the input pattern of an exemplar are interpreted as being equally important in solving the classification problem, but this is not necessarily the case. We therefore weigh each feature with its *information gain*; a number expressing the relevance of the feature in terms of the average amount of reduction of information entropy in memory when knowing the value of the feature (Daelemans & van den Bosch, 1992, Quinlan, 1993; Hunt et al. 1966) (Equation 3). We will call this algorithm IB1-IG. Many other methods to weigh the relative importance of features have been designed, both in statistical pattern recognition and in machine learning (see Wettschereck et al. 1996 for an overview).

The main idea of *information gain weighting* is to interpret the memory exemplars as an information source capable of generating a number of messages (the different output category labels) with a certain probability. The information entropy of such an information source can be compared in turn for each feature to the average information entropy of the information source when the value of that feature is known. Database information entropy is equal to the number of bits of information needed to know the category given a pattern, where the probability of a category is estimated by its relative frequency in memory. From this amount, for each feature, the average memory information entropy when knowing each of the values of that feature, is subtracted (Equation 3).

$$\begin{aligned} w(f) \quad = \quad & -\sum_C P(C) \log_2 P(C) \\ & -\sum_V \left(-P(V_f) \times \sum_C P(C|V_f) \log_2 P(C|V_f)\right) \end{aligned} \tag{3}$$

Finally, the distance metric in Equation 1 is modified to take into account the information gain weight associated with each feature (Equation 4).

$$\Delta(X, Y) = \sum_{i=1}^{n} w(f_i)\delta(x_i, y_i) \tag{4}$$

It is important to emphasize that the metrics used in exemplar-based models are, and should be, *language-independent*: they are not linguistically informed, and are applicable to domains as different as perception, medical diagnosis and robotics. This is crucial, because otherwise, the nativist presuppositions would move from the universal grammar principles and rules to the tailoring of linguistically motivated metrics. The *representations* on which these metrics work (i.e. the feature-value inventory used to describe the input and output parts of the exemplars) *are* of course linguistically informed; they are linguistic representations. Ideally, this feature-value inventory (the ontology used by the representations) should be derived automatically from the data as well, a problem investigated with *unsupervised learning* approaches. Progress has been made in these areas. See e.g. Zavrel & Veenstra (1996) for an example of the distributional bootstrapping of syntactic categories, and Daelemans et al. (1996) where it is shown how learning to produce morphologically complex words allows a system to discover phonological features automatically. We will presuppose the existence of the feature-inventories of our linguistic representations in what follows, however.

4 Experimental Results

In previous research[1], we have applied the experience-driven approach to a number of language processing tasks: *syllabification* (segment a word into syllables taking into account morphological structure), *grapheme-to-phoneme conversion* (identify the pronunciation of words), *stress assignment* (identify the stress pattern of words), *morphology* (both synthesis and analysis), and *morphosyntactic disambiguation* (identify for each word in a text its morpho-syntactic category). See Daelemans (1995) for a discussion of the general approach, and van den Bosch & Daelemans, 1992, 1993; Daelemans & van den Bosch, 1992ab, 1993, 1994; van den Bosch et al. 1996; Daelemans et al. 1994, 1995, 1996ab for the details.

There are some general trends which become clear when analysing the results of all these experiments. First, the most striking result is that the generalization accuracy of the induced systems is always comparable and often better than equivalent hand-crafted systems. Second, when comparing different non-abstracting experience-driven algorithms (notably IB1, IB1-IG, and variants) to other learning approaches which *do* abstract from the data (the decision tree and rule induction programme C4.5 or Backprop learning in connectionist networks, e.g.), we find that IB1-IG (the simplest possible exemplar-based algorithm, extended with information-entropy-based feature weighting and a probabilistic decision rule) *always* obtains the best generalization accuracy. The picture is less clear for second place. Thirdly, in the same comparison, there is a tendency that the more abstracting the technique is, the less accurate generalization becomes. More theoretical and empirical work is needed to explain and refine these results. However, it is clear that the universal structure of linguistic tasks (some regularities, many subregularities and many exceptions) favours a learning approach in which abstraction from specific exemplars is avoided.

We will briefly describe morphosyntactic disambiguation as an example of the experience-driven approach (Daelemans et al. 1996 for details) and show how a complete syntactic analysis approach can be based on a similar approach.

[1] With Peter Berck, Antal van den Bosch, Gert Durieux, Steven Gillis, Ton Weijters, and Jakub Zavrel.

4.1 An Illustration: Morphosyntactic Disambiguation

The problem of morphosyntactic desambiguation (for a human language understander as well as an automatic one) is the following: given a text, provide for each word in the text its contextually disambiguated morphosyntactic category. I.e., the words in the sentence `the old man the boats` should be interpreted as belonging to the following parts of speech respectively: `Art Noun Verb Art Noun`. In experiments on automatic morphosyntactic disambiguation, the inventory of part of speech categories ranges from extremely simple (order 10) to extremely complex (order 1000). Ideally, the inventory should be learned from the data in an unsupervised way, e.g. through distributional clustering. Morphosyntactic disambiguation is a hard task because of the massive ambiguity in natural language text. E.g. in the example above, `man` can be both a noun and a verb, lexical probability forces at first a noun reading, but context determines that in this case it is a verb. The correct category of a word thus depends on a smooth integration of its lexical probability $Pr(cat|word)$, and its contextual probability $Pr(cat|context)$.

There are rule-based systems (hand made or using rule-induction), and statistical systems (using markov modeling and dynamic programming) to solve this task. Although a thorough and reliable comparison of these approaches has not yet been achieved, it seems to be the case that all approaches converge to a 96-97% accuracy on new text from the same type as the training material.

The architecture of our exemplar-based morphosyntactic disambiguator takes the following form: given a corpus tagged with the desired morphosyntactic categories (the experience of the system), a disambiguator is produced which maps the words of new text to categories according to the same systematicity. This is achieved in the following way. Given the annotated training corpus, three datastructures are automatically extracted: a *lexicon* (associating words to possible categories as evidenced in the training corpus), an exemplar memory for *known words* (words occurring in the lexicon), and an exemplar memory for *unknown words*. During disambiguation, each word in the text is looked up in the lexicon. If it is found, its lexical representation is retrieved and its context is determined, and the resulting pattern is disambiguated using extrapolation from the most similar exemplars in the known words memory. When a word is not found in the lexicon, its lexical representation is computed on the basis of its form, its context is determined, and the resulting pattern is disambiguated using extrapolation from the most similar exemplars in the unknown words exemplar memory. In each case, output is a best guess of the category for the word in its current context.

For known words, exemplars consist of information about a focus word to be disambiguated, its left and right context, and an associated category valid for the focus word in that context. For unknown words, a category can be guessed only on the basis of the *form* or the *context* of the word. In our experience-driven approach, we provide word form information (especially about suffixes) indirectly to the disambiguator by encoding the three last letters of the word as separate features in the exemplar representation. The first letter is encoded as well because it contains information about prefix and capitalization of the word. Context information is added to the exemplar representation in a similar way as with known words.

For evaluation, we performed the complete disambiguator generation process on a 2 million words training set (lexicon construction and known and unknown words exemplar memory construction), and tested on 200,000 test words. Generalization performance on known words (96.7%), unknown words (90.6%), and total (96.4%) is competitive with alternative hand-crafted and statistical approaches. The experiment shows that it is possible to achieve high accuracy morphosyntactic disambiguation in a system that does not abstract on the basis of experiences, but rather uses the exemplars themselves directly.

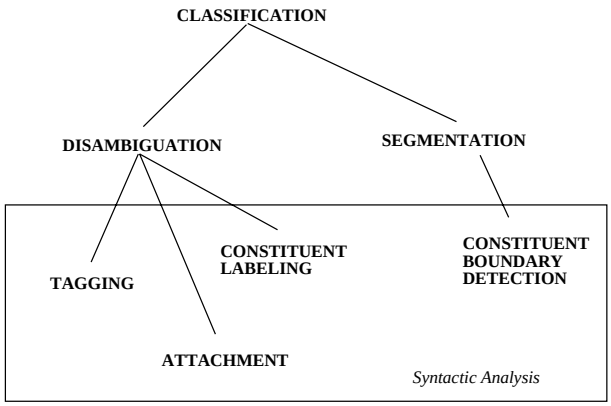
4.2 Syntax

Exemplar-based acquisition and processing is fundamentally a *classification* paradigm. Given a description in terms of feature-value pairs of an input, an output category is produced. This category should normally be taken from a finite inventory of possibilities, known beforehand. How could such a paradigm be used for syntactic analysis?

It is our claim that all useful linguistic tasks, including parsing, can be redefined as classification. All linguistic problems can be described as context-sensitive mappings. These mappings can be of two kinds: *disambiguation* and *segmentation* (disambiguation of boundaries) (see Daelemans, 1995).

- **Disambiguation**. Given a set of possibilities (categories) and a relevant context in terms of attribute values, determine the correct possibility for this context. Instances of identification include *morphosyntactic disambiguation*, *grapheme-to-phoneme conversion*, *lexical selection in generation*, *morphological synthesis*, *word sense disambiguation*, *term translation*, *stress assignment*, etc.
- **Segmentation**. Given a target and a context, determine whether a boundary is associated with this target, and if so which one. Examples include *syllabification*, *morphological analysis*, *syntactic analysis*, etc.

Syntactic analysis (parsing) could then be defined as a cascade of morphosyntactic disambiguation (tagging), segmentation into constituents, constituent disambiguation (labeling), and disambiguation of relations between constituents (attachment).



It remains to be seen whether each of these in itself plausible subproblems (many of which have already been solved in an experience-driven framework), produce useful syntactic analyses when combined. Preliminary results on some segmentation problems (finding NPs in text) and some attacthment problems (finding the correct attachment point for PPs) are encouraging.

5 Conclusion

We presented an exemplar-based model of language acquisition and processing compatible with but more radical than cognitive grammar. The model is performance-oriented and is based on the storage of task-dependent input-output patterns (exemplars) in memory. Acquisition consists of filling memory incrementally with such patterns, and processing is based on retrieval of previously stored exemplars, and similarity-based (analogical) extrapolation from previously stored exemplars to new task instances. We discussed the implementation of a computational model for this theoretical framework in which similarity is defined by means of a task- and language-independent distance metric based

on work in information theory and statistical pattern recognition. Several experiments on phonological, morphological, and morphosyntactic linguistic performance tasks show that the generalization accuracy of this simple computational model (its ability to solve previously unencountered cases) is comparable to or better than hand-crafted or induced rule-based approaches, suggesting that abstraction is harmful in learning linguistic performance tasks. Finally, we showed how such an exemplar-based model could be applied to the more complex problem of syntactic analysis.

6 References

Aha, D. W., Kibler, D., & Albert, M. (1991). 'Instance-based learning algorithms'. *Machine Learning*, 7, 37–66.

van den Bosch, A. and W. Daelemans. (1992). Linguistic Pattern Matching Capabilities of Connectionist Networks. In: Jan van Eijck and Wilfried Meyer Viol (Eds.) *Computational Linguistics in the Netherlands. Papers from the Second CLIN-meeting 1991.* Utrecht: OTS, 40-53.

Van den Bosch, A. and Daelemans, W. (1993). 'Data-oriented methods for grapheme-to-phoneme conversion.' Proceedings of the Sixth conference of the European chapter of the ACL, ACL, 45–53.

Van den Bosch, A., W, Daelemans, T. Weijters. (1996). 'Morphological Analysis as Classification: an Inductive-Learning Approach.' *Proceedings of NEMLAP 1996*, Ankara, Turkey.

Cardie, C. (1994). 'Domain-Specific Knowledge Acquisition for Conceptual Sentence Analysis'. Ph.D. Thesis, University of Massachusetts, Amherst, MA.

Chandler, S. (1992). 'Are rules and modules really necessary for explaining language?' *Journal of Psycholinguistic research*, 22(6): 593–606.

Daelemans, W. (1995). 'Memory-based lexical acquisition and processing.' In Steffens, P., editor, *Machine Translation and the Lexicon*, Lecture Notes in Artificial Intelligence 898. Berlin: Springer, 85–98.

Daelemans, W., Van den Bosch, A. (1992a). 'Generalisation performance of backpropagation learning on a syllabification task.' In M. Drossaers & A. Nijholt (Eds.), *TWLT3: Connectionism and Natural Language Processing.* Enschede: Twente University, 27–38.

Daelemans, W. and A. van den Bosch. (1992b)'A Neural Network for Hyphenation.' In: I. Aleksander and J. Taylor (eds.) *Artificial Neural Networks II: Proceedings of the International Conference on Artificial Neural Networks.*, Elsevier Science Publishers, 1647-1650.

Daelemans, W. and A. van den Bosch. (1993). 'TABTALK: Reusability in Data-oriented grapheme-to-phoneme conversion.' *Proceedings of Eurospeech*, Berlin, 1459-1466.

Daelemans, W. and A. van den Bosch. (1994). 'A language-independent, data-oriented architecture for grapheme-to-phoneme conversion.' In: *Proceedings of the ESCA-IEEE conference on Speech Synthesis*, New York, 199-203.

Daelemans, W., S. Gillis and G. Durieux. (1994). 'The Acquisition of Stress, a data-oriented approach.' *Computational Linguistics* 20 (3), 421-451.

Daelemans, W., P. Berck, and S. Gillis. (1995). 'Linguistics as Data Mining: Dutch Diminutives', in Andernach, T., M. Moll, and A. Nijholt (eds). *CLIN V, Papers from the Fifth CLIN Meeting*, 59-72.

Daelemans, W., J. Zavrel, P. Berck, S. Gillis. (1996a). 'MBT: A Memory-Based Part of Speech Tagger-Generator'. In: E. Ejerhed and I. Dagan (eds.) *Proceedings of the Fourth Workshop on Very Large Corpora*, Copenhagen, Denmark, 14-27.

Daelemans, W., P. Berck, S. Gillis. (1996b).'Unsupervised Discovery of Phonological Categories through Supervised Learning of Morphological Rules.' *Proceedings of the 16th International Conference on Computational Linguistics* (COLING-96), Copenhagen, Denmark, 95-100.

Derwing, B. L. and Skousen, R. (1989). 'Real Time Morphology: Symbolic Rules or Analogical Networks'. *Berkeley Linguistic Society* 15: 48–62.

Federici S. and V. Pirelli. (1996). 'Analogy, Computation and Linguistic Theory.' In Jones, D. (ed.) *New Methods in Language Processing.* London: UCL Press, forthcoming.

Friedman, J., Bentley, J., and Ari Finkel, R. (1977). 'An algorithm for finding best matches in logarithmic expected time.' *ACM Transactions on Mathematical Software*, 3(3), 209–227.

Hunt, E., J. Marin, P. Stone. (1966). *Experiments in Induction.* New York: Academic Press.

Jones, D. *Analogical Natural Language Processing.* (1996). London: UCL Press.

Kolodner, J. (1993). *Case-Based Reasoning.* San Mateo: Morgan Kaufmann.

Langacker, R. (1991). *Concept, Image, and Symbol. The Cognitive Basis of Grammar.* Berlin: Mouton De Gruyter.

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

Salzberg, S. (1990) 'A nearest hyperrectangle learning method'. *Machine Learning* 6, 251–276.

Scha, R. (1992) 'Virtuele Grammatica's en Creatieve Algoritmen.' *Gramma/TTT* 1 (1), 57–77.

Skousen, R. (1989). *Analogical Modeling of Language.* Dordrecht: Kluwer.

Stanfill, C. and Waltz, D. (1986). 'Toward memory-based reasoning.' *Communications of the ACM*, 29, 1212–1228.

Wettschereck, D., Aha, D.W. & Mohri, T. (1996). 'A review and comparative evaluation of feature weighting methods for lazy learning algorithms' Technical Report AIC-95-012. Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Zavrel, J. and J. Veenstra. (1996) 'The Language Environment and Syntactic Word-Class Acquisition', in C.Koster & F.Wijnen (eds.), *Proc. of the Groningen Assembly on Language Acquisition* (GALA95), University of Groningen.