A Comparison of Analogical Modeling of Language to Memory-Based Language Processing

Walter Daelemans*
CNTS, Language Technology Group, University of Antwerp
ILK Research Group, Tilburg University

DRAFT, August 2000

Abstract

Memory-Based Language Processing (MBLP), like Analogical Modeling of Language (AML), is an approach to modeling language learning and language processing that is based on the idea that language behavior is guided by the direct reuse of memory traces of earlier language experience rather than by rules extracted from such experience. Despite their similarities, both approaches show important theoretical, algorithmic, and empirical differences. MBLP uses algorithms and metrics taken from statistical pattern recognition (nearest neighbor methods), and information theory. AML is based on a natural (psychologically plausible) statistic. We will discuss these differences, focusing on new empirical work comparing AML and MBLP on learning and processing plural formation in German.

1 Introduction

Memory-Based Language Processing is inspired by the hypothesis that in learning a cognitive task from experience, people do not extract rules or other abstract representations from their experience, but reuse their memory of that experience directly. For language behavior modeling, this means that *language acquisition* is simply the storage of experiences in memory, and *language processing* is the result of analogical reasoning on memory structures. Whereas the inspiration and motivation for our approach to MBLP has come mainly from statistical pattern recognition, and Artificial Intelligence, a similar approach has also survived the Chomskyan revolution in linguistics, most notably in the work of Royal Skousen on Analogical Modeling of Language. After presenting a short history and characterisation of both MBLP and AML in this section, we will discuss the main algorithmic differences in Section 2, and study their effects empirically in Section 3 in a comparative study using the German plural as a benchmark task. Section 4 discusses theoretical implications of the emprirical results.

1.1 Memory-Based Language Processing

As far as the algorithms used in MBLP are concerned, nearest neighbor methods ($k-nn$), developed in statistical pattern recognition from the fifties onwards, have played an important inspirational role (Fix and Hodges, 1951; Cover and Hart, 1967). In these methods,

*Research partially supported by FWO (Belgium) and NWO (The Netherlands). Many thanks to the members of ILK and CNTS for providing inspiring working environments, and to the participants to the AML conference for useful discussion and comments. Special thanks to Gert Durieux for sharing his expertise about, and implementation of, AML, and for help with preprocessing the CELEX data.

examples (labeled with their class) are represented as points in an example space with as dimensions the numeric attributes used to describe them. A new example obtains its class by finding its position as a point in this space, and extrapolating its class from the k nearest points in its neighborhood. Nearness is defined as the reverse of Euclidean distance. This literature has also generated many studies on methods for removing examples from memory either for efficiency (faster processing by removing unnecessary examples) or for accuracy (better predictions for unseen cases by removing badly predicting examples). See Dasarathy (1991) for a collection of fundamental papers on $k-nn$ research. However, until the nineteen eighties, the impact of these non-parametric statistical methods on the development of systems for solving practical problems has remained limited because of a number of shortcomings: they were computationally expensive in storage and processing; intolerant of attribute noise and irrelevant attributes; sensitive to the similarity metric used; and the Euclidean distance metaphor for similarity breaks down with non-numeric and missing feature values.

From the late eighties onwards, the intuitive appeal of the nearest neighbor approach has been adopted in Artificial Intelligence in many variations on the basic nearest neighbor modeling idea, using names such as memory-based reasoning, case-based reasoning, exemplar-based learning, locally-weighted learning, and instance-based learning (Stanfill and Waltz, 1986; Cost and Salzberg, 1993; Riesbeck and Schank, 1989; Kolodner, 1993; Atkeson, Moore, and Schaal, 1997; Aamodt and Plaza, 1994; Aha, Kibler, and Albert, 1991). These methods modify or extend the nearest neighbor algorithm in different ways, and aim to solve (some of) the problems with $k-nn$ listed before. Recently, the term *Lazy Learning* (as opposed to *Eager Learning*) has been proposed as a generic term for this family of methods (Aha, 1997).

Since the early nineteen nineties, we find several studies using nearest-neighbor techniques for solving problems in Natural Language processing (Cardie, 1996; Daelemans, Van den Bosch, and Zavrel, 1999). The general approach is to define the tasks as (cascades of) classification problems. For each (sub)problem instances are collected of input linguistic items and their context, and an associated output linguistic class. The German plural prediction task to be discussed later adheres to this format. The spectrum of language processing tasks that has been investigated within this framework ranges from phonology to semantics and discourse processing. See (Daelemans, 1999) for a recent overview.

A related framework is DOP (Data-Oriented Parsing), a memory-based approach to syntactic parsing (Scha, Bod, and Sima'an, 1999), which uses a corpus of parsed or semantically analyzed utterances (a treebank) as a representation of a person's language experience, and analyzes new sentences searching for a recombination of subtrees that can be extracted from this treebank. The frequencies of these subtrees in the corpus are used to compute the probability of analyses.

In another related tradition, Nagao (1984) proposed Example-Based Machine Translation (EBMT), an approach to Machine Translation which is essentially memory-based. By storing a large set of (analyzed) sentences or sentence fragments in the source language with their associated translation in the target language as exemplars, a new source language sentence can be translated by finding exemplars in memory that are similar to it in terms of syntactic structure and word meaning, and extrapolating from the translations associated with these examples. Especially in the UK and Japan, this approach has become an important subdiscipline within Machine Translation research.

1.2 Analogical Modeling of Language

Since Chomsky replaced the vague notions of analogy and induction existing in linguistics in his time (in work of e.g., de Saussure and Bloomfield) by the clearer and better operationalised notion of rule-based grammars, most mainstream linguistic theories, even

the functionally and cognitively inspired ones, have assumed rules to be the only or main means to describe any aspect of language.

In contrast, Royal Skousen (1989; 1992) argues for a specific operationalisation of the pre-Chomskyan analogical approach to language and language learning (AML). He introduced a definition of analogy that is not based on rules and that does not make a distinction between regular instances (obeying the rules) and irregular instances (exceptions to the rules). To model language acquisition and processing, a database of examples of language use is searched looking for instances analogous to a new item, and extrapolating a decision for the new item from them.

Current research on AML attempts to solve the computational complexity problem (the algorithm is exponential in the number of attributes used to describe examples), and to apply the approach to a wide range of linguistic problems. The work has also been taken up as a psycholinguistically relevant explanation of human language acquisition and processing, especially as an alternative to *dual route* models of language processing (Eddington, 2000; Chandler, 1992; Derwing and Skousen, 1989). AML has also been used in computational linguistics. Jones (1996) describes an application of AML in Machine Translation, and work by Deryle Lonsdale includes AML implementations of part-of-speech tagging and sentence boundary detection.

While AML is the most salient example of analogy-based theories in linguistics (and the most interesting from a computational point of view), other linguists outside the mainstream have proposed similar ideas. E.g., in the storage versus computation trade-off in models of linguistic processing, linguists like Bybee (1988), and usage-based linguistic theories such as Cognitive Grammar (Langacker, 1991) claim an important role for examples (instances of language use); but they still presuppose rules to be essential for representing generalizations.

2 A Comparison of Algorithms

Whereas AML refers to a single algorithm, there are various possible ways in which ideas in MBLP can be operationalized in algorithmic form. In the remainder of this text, we will narrow down our discussion of MBLP to the specific incarnation of it that has been used intensively in Tilburg and Antwerp. Although our specific approach to MBLP was developed primarily with language engineering purposes in mind, like in AML, its linguistic and psycholinguistic relevance has always been a focus of attention. As an example, work on word stress acquisition and processing in Dutch contrasted MBLP with metrical phonology and studied correlations between errors made by a memory-based learner and those made by children producing word stress in a repetition task (Daelemans, Gillis, and Durieux, 1994; Steven Gillis, 2000). Many of the properties which make AML cognitively and linguistically plausible also apply to MBLP: (i) there is no all-or-none distinction between regular cases and irregular cases because no rules are used, (ii) fuzzy boundaries and leakage between categories occurs, (iii) the combination of memory storage and similarity-based reasoning is cognitively simpler than rule-discovery and rule processing, and (iv) memory-based systems show adaptability and robustness. Remarkably, seen from the outside, such analogical or memory-based approaches appear to be rule-governed, and therefore adequately explain linguistic intuitions as well.

Both approaches are instances of the same general view of cognitive architecture. However, because of the different algorithm used to extrapolate outcomes from stored occurrences, the properties and behavior of both approaches may differ considerably in specific cases.

2.1 Similarity in MBLP

The most basic metric that works for patterns with symbolic features as well as for numeric features, is the *overlap metric* given in equations 1 and 2; where $\Delta(X, Y)$ is the distance between patterns X and Y, represented by n features, and δ is the distance per feature. The distance between two patterns is simply the sum of the differences between the features. The $k-nn$ algorithm with this metric is called IB1[1] (Aha, Kibler, and Albert, 1991). Usually k is set to 1.

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

where:

$$\delta(x_i, y_i) = \begin{cases} \frac{x_i - y_i}{max_i - min_i} & \text{if numeric, else} \\ 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \tag{2}$$

The distance metric in equation 2 simply counts the number of (mis)matching feature-values in both patterns. In the absence of information about feature relevance, this is a reasonable choice. However, we can do better by computing statistics about the relevance of features by looking at which features are good predictors of the class labels. Information Theory gives us a useful tool for measuring feature relevance in this way. *Information Gain* (IG) weighting looks at each feature in isolation, and measures how much information it contributes to our knowledge of the correct class label (Quinlan, 1993). The Information Gain of feature i is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature (equation 3).

$$w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C|v) \tag{3}$$

Where C is the set of class labels, V_i is the set of values for feature i, and $H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$ is the entropy of the class labels. The probabilities are estimated from relative frequencies in the training set. For numeric features, values are first discretized into a number (the default is 20) of equally spaced intervals between the minimum and maximum values of the feature. These groups are then used in the IG computation as if they were discrete values (note that this discretization is not used in the computation of the distance metric). The $k-nn$ algorithm with this metric is called IB1-IG (Daelemans and van den Bosch, 1992). For more references and information about the algorithms we refer to Daelemans, Van den Bosch, and Weijters (1997; Daelemans et al. (1998; Daelemans, Van den Bosch, and Zavrel (1999).

For most of our experiments in the past, IB-IG with extrapolation based on 1 nearest neighbor ($k = 1$) has been the default MBLP algorithm. Note that setting $k = 1$ may imply extrapolation from more than one exemplar in memory: in case there is more than one exemplar which is the nearest neighbor, the algorithm uses all of them for extrapolation, and selects the class which appears most in them (or the overall most frequent class in case of ties). In what follows, we will use both IB1-IG (the particular incarnation) and MBLP (the general approach) to refer to our approach, depending on the context.

[1]For our experiments we have used TiMBL, available from `http://ilk.kub.nl/`. It is a Memory-Based Learning software package developed in our group (Daelemans et al., 1998). TiMBL implements a number of important memory-based algorithms and metrics. We only describe those here which we used in the experiments below.

2.2 The AML Extrapolation Algorithm

The main algorithmic difference between AML and MBLP is the way the selection of memory items to extrapolate from is made. In IB1-IG, the different features are assigned a relative importance, which is used during matching to filter out the influence of irrelevant features. In AML, essentially the same effect is achieved without precomputing the relative importance of individual features[2]. Instead, all features are equally important initially, and serve to partition the database into several disjoint sets of examples. Filtering out irrelevant exemplars is done by considering properties of these sets rather than by inspecting individual features that their members may share with the input pattern. To explain how this works, we will describe the matching procedure in more detail[3].

The first stage in the matching process is the construction of *subcontexts*; subcontexts are sets of examples, and they are obtained by matching the input pattern, feature by feature, to each item in the database, on an equal/not equal basis, and classifying the database exemplars accordingly. Taking an input pattern ABC as an example, eight (2^3) different subcontexts would be constructed, ABC, $\overline{A}BC$,$A\overline{B}C$, $AB\overline{C}$, $\overline{AB}C$, $\overline{A}B\overline{C}$, $A\overline{BC}$ and $\overline{ABC}$, where the overstrike denotes complementation. Thus, exemplars in the class ABC share all their features with the input pattern, whereas for those in $\overline{AB}C$ only the value for the third feature is shared. In general, n features yield 2^n mutually disjoint subcontexts. Subcontexts can be either *deterministic*, which means that their members all have the same associated category, or *non-deterministic*, when several categories occur.

In the following stage, *supracontexts* are constructed by generalising over specific feature values. This is done by systematically discarding features from the input pattern, and taking the union of the subcontexts that are subsumed by this new pattern. Supracontexts can be ordered with respect to generality, so that the most specific supracontext contains examples which share all n features with the input pattern, less specific supracontexts contain items which share at least $n-1$ features, and the most general supracontext contains all database exemplars, whether or not they have any features in common with the input pattern. In the table below the supracontexts for our previous example are displayed, together with the subcontexts they subsume.

Supracontext			Subcontexts
A	B	C	ABC
A	B	-	ABC $AB\overline{C}$
A	-	C	ABC $A\overline{B}C$
-	B	C	ABC $\overline{A}BC$
A	-	-	ABC $A\overline{B}C$ $AB\overline{C}$ $A\overline{BC}$
-	B	-	ABC $\overline{A}BC$ $AB\overline{C}$ $\overline{AB}\overline{C}$
-	-	C	ABC $\overline{A}BC$ $A\overline{B}C$ $\overline{AB}C$
-	-	-	ABC $\overline{A}BC$ $A\overline{B}C$ $AB\overline{C}$ $\overline{AB}C$ $\overline{A}B\overline{C}$ $A\overline{BC}$ $\overline{ABC}$

An important notion with respect to supracontexts is *homogeneity*. A supracontext is called homogeneous when any of the following conditions holds:

- The supracontext contains nothing but empty subcontexts.
- The supracontext contains only deterministic subcontexts with the same category.
- The supracontext contains a single non-empty, non-deterministic subcontext.

[2]The specific analogical algorithm employed by Skousen is available in a number of implementations. See the AML group's homepage at `http://humanities.byu.edu/aml/homepage.html`. For our experiments, we used an implementation by Gert Durieux, AML 0.1, available from `durieux@ua.ac.be`.

[3]This description of the algorithm was taken from Daelemans, Gillis, and Durieux (1997).

Heterogeneous supracontexts are obtained by combining deterministic and non-deterministic subcontexts. Going from least to most general, this means that as soon as a supracontext is heterogeneous, any more general supracontext will be heterogeneous too.

In the final stage, the analogical set is constructed. This set contains all of the exemplars from each of the homogeneous supracontexts. Two remarks are in order here. First, since some exemplars will occur in more than one supracontext, each exemplar is weighted according to its distribution across different supracontexts. This is accomplished by maintaining a score for each exemplar. This score is simply the summed cardinality of each of the supracontexts in which the exemplar occurs. The motivation for this scoring mechanism is to favor frequent patterns over less frequent ones, and patterns closer to the input pattern over more distant patterns, since the former will surface in more than one supracontext. Second, banning heterogeneous supracontexts from the analogical set ensures that the process of adding increasingly dissimilar exemplars is halted as soon as those differences may cause a shift in category. Exactly when this happens depends largely on the input pattern.

To finally categorise the input pattern, either the predominant category in the analogical set (plurality) or the category of a probabilistically chosen member of this set is chosen.

2.3 AML versus MBLP

The different way in which IB1-IG and AML construct a set of exemplars to extrapolate from, leads to a number of differences which have sometimes been advanced as an advantage or disadvage for one or the other approach (Skousen, 2000; Daelemans, Gillis, and Durieux, 1997). We will list these differences here, and discuss them in the context of our experimental results in Section 4.

1. Non-neighbors can affect language behavior in AML, not in IB1-IG.
2. Because of the method of constructing contexts, AML can locally determine the significance of variables (feature values), whereas these are lost in the averaging over values when using information gain in IB1-IG.
3. The feature weighting in IB1-IG constitutes a type of preprocessing or learning which is unnecessary in AML.
4. The natural statistic on which AML is based makes necessary the use of only a percentage of the data (imperfect memory) for optimal accuracy and robustness, whereas for IB1-IG "forgetting exceptions is harmful to language learning" (Daelemans, Van den Bosch, and Zavrel, 1999).
5. AML is exponential in the number of features, IB1-IG is linear in the number of features and in the number of exemplars.
6. AML has no natural extension to numeric data whereas the overlap metric used in IB1-IG can be easily generalized to different types of feature values (numeric, set-valued).

3 Empirical Comparison: German Plural

The diachrony of plural formation of German nouns has led to a notoriously difficult system, which is nevertheless routinely acquired by speakers of German. Because of the complex interaction, from a synchronic point of view, of regularities, subregularities, and exceptions, it is to be expected that lexicon-based methods such as AML and IB1-IG do well in this case, and that it is an interesting testing ground for comparing them.

There is another reason why the German plural is an interesting problem. Marcus et al. (Clahsen, 1999; Marcus et al., 1995) have argued that this task provides evidence for

the *dual route* model for cognitive architectures. A dual route architecture supposes the existence of a cognitively real productive mental default rule, and an associative memory for irregular cases which blocks the application of the default rule. They argue that $-s$ is the regular plural in German, as this is the suffix used in many conditions associated with regular inflection (e.g. novel words, surnames, acronyms, etc.). This default rule is applied whenever associative memory-lookup fails. The case of German plurals provides an interesting new perspective to what is *regular*: in this case, the default rule (regular route) is less frequent than many of the 'irregular' associative memory cases. In a plural noun suffix type frequency ranking (see below), $-s$ comes only in last place. Perhaps the behavior of AML and IB1-IG as *single route* models offers some additional insight into this phenomenon.

We collected 25,753 German nouns from the German part of the CELEX-2 lexical database[4]. We removed from this dataset cases without plurality marking, cases with Latin plural in $-a$, and a miscellaneous class of foreign plurals. From the remaining 25,168 cases, we extracted or computed for each word the plural suffix, the gender feature, and the syllable structure of the two last syllables of the word in terms of onsets, nuclei, and codas expressed using a phonetic segmental alphabet. Table 1 gives an overview of the features, values, and output classes considered in these experiments. The gender feature has, apart from masculin (M), neutre (N), and feminine (F) also all possible combinations of two genders.

Table 1: Data characteristics of German Plural experiments.

Feature	number of values	Example: *Vorlesung*
Onset penultimate	78	l
Nucleus penultimate	27	e
Coda penultimate	85	-
Onset last	84	z
Nucleus last	27	U
Coda last	79	N
Gender	10	F
Class	8	-en

Table 2 lists the possible output classes with their type frequency in the dataset. There was no further preprocessing of the data. A well-known source of *noise* in the CELEX data are plain mistakes in lexical coding. However, we expect learning methods to be robust to this type of noise, and did not attempt to find and correct these coding errors.

In order to empirically compare the accuracy of AML and IB1-IG on the German plural task, we performed several learning experiments. We compared the learnability of the task varying the training set size for the complete task and for the different suffixes separately, we performed an error analysis and comparison, and we looked at the influence of some different parameter settings on algorithm accuracy.

3.1 Learnability

In an initial learnability experiment, we randomized the dataset, selected a 5,168 word test set, and divided the remaining 20,000 words in 19 training sets with an incrementally

[4]Available from `http://www.ldc.upenn.edu/`

Table 2: Type frequency of pluralization mechanisms in Celex.

Class	Frequency	Umlaut	Frequency	Example
(e)n	11920			Abart
e	6656	no	4646	Abbau
		yes	2010	Abdampf
-	4651	no	4402	Aasgeier
		yes	249	Abwasser
er	974	no	287	Abbild
		yes	687	Abgang
s	967			Abonnement

increasing size from 200 to 2,000 in steps of 200, and from 2000 up to 20,000 in steps of 2,000. Each of the algorithms was then trained with each of the training sets and tested each time on the single test set. Figure 3.1 shows the learning curve for both algorithms when using their standard settings, i.e. IB1-IG with information gain and $k = 1$ for MBLP, and AML with perfect memory and with plurality selection.

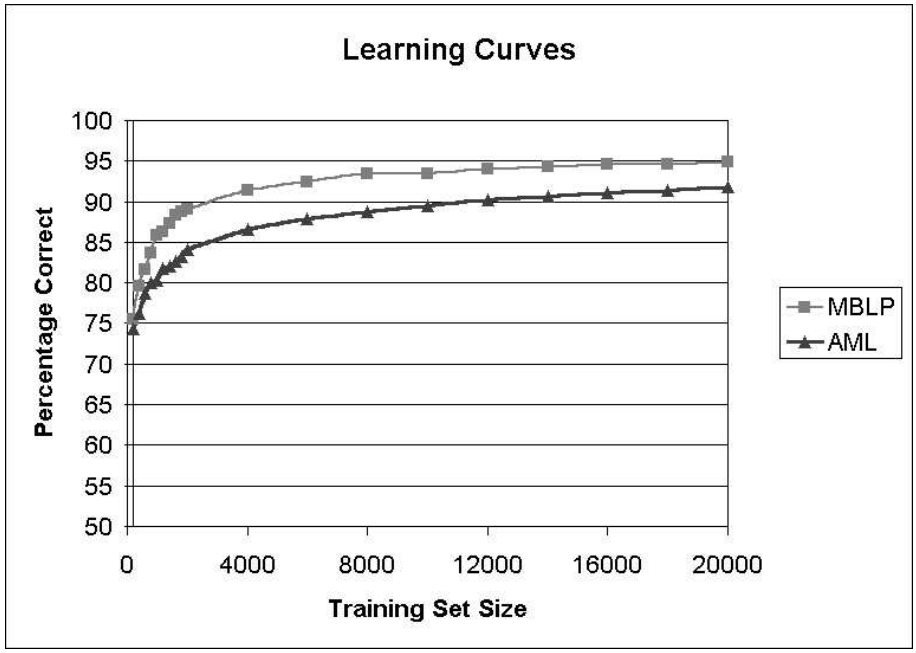

Figure 1: *Learnability of German Plural with* MBLP *and* AML.

We see that for small training sets, AML performs about the same as MBLP, but a statistically significant divergence in favor of MBLP starts after 1000 training items. Although accuracy is still increasing for both algorithms with 20,000 training cases, learning seems to come near to its upper bound already at around 2,000 training cases.

In Figures 3.1 and 3.1 the learning curves of the individual plural formation classes are shown for AML and IB1-IG, respectively[5]. Interestingly, for both algorithms, the suffixes seem to fall into three classes: those that are learned correctly from the start (*−en* and

[5]In these figures, the training set sizes on the x-axis are represented as categorical values, i.e., the 200 item

$-$), those that require longer learning but are learned very well in the end ($-e$ and $-er$), and one which is never learned very well at all ($-s$), although accuracy is increasing with number of training items. It seems indeed to be the case that $-s$ behaves differently from the other suffixes, when learned by single-route models such as AML and IB1-IG. However, this does not necessarily lend credence to a *dual route* model for the German plural. The learning curves clearly show that the suffix is learned by single route models as well (at least some generalizations about when to use $-s$ are learned), and 60% accuracy (for IB1-IG) is a respectable result given the limited information provided in the input representations. It is by no means inconceivable that additional semantic of syntactic features could further improve learnability of $-s$ with the single route models discussed here. The only conclusion that can be drawn from these experiments in this regard is that whereas the other suffixes are learnable from syllable structure and gender information, this is not the case for $-s$.

For those suffixes which are sometimes accompanied by an Umlaut, there is no marked difference in the speed of learning and accuracy achieved for versions with and without Umlaut. For the different suffixes, we see that AML learning is slower and reaches lower accuracies, except for the $-en$ suffix shich is learned very well from the start by AML.

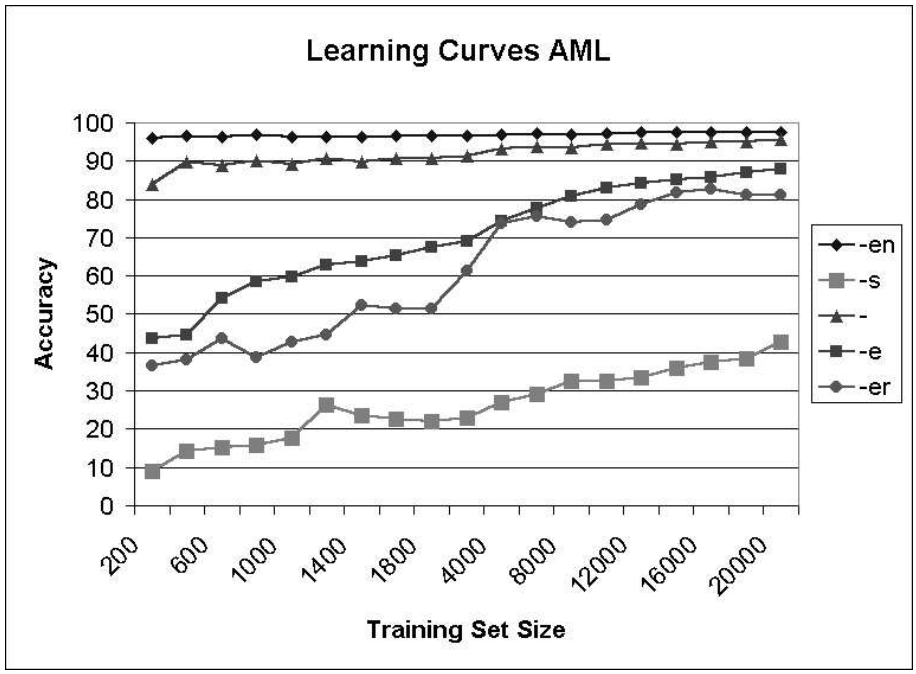

Figure 2: *Learnability of German Plural classes with* AML

3.2 Error Analysis

In order to generate more data for a comparison between AML and MBLP on the German plural data, we performed a leave-one-out experiment using both algorithms. In such a set-up, each instance in the data file is held out in turn as a test item, and all remaining instances act as training material to train the classifier. In machine learning methodology, the leave-one-out method is generally accepted as the best estimator for the "real" error

training sets get as much space on the x-axis as the 2000 item datasets, hence the less steep learning curve, compared to Figure 3.1.

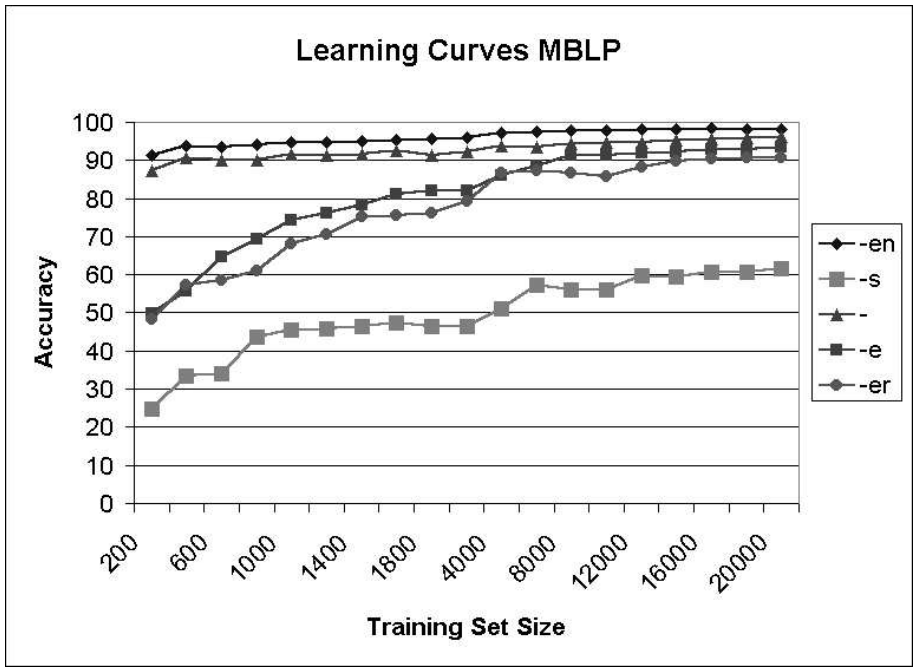

Figure 3: *Learnability of German Plural classes with* MBLP

of a classifier. The advantage of using it in this context, is that we have access to the complete dataset to look for trends or examples. For both algorithms, we again used the default parameter settings. Table 3 shows the accuracy on the full dataset using this method for AML and MBLP distributed over the different suffixes. For clarity, we repeated the frequencies of Table 2 above.

The high accuracies found in both algorithms are partly due to exact matches in memory: several different words can have the same syllable structure for their last two syllables and the same gender. Disregarding these cases (i.e. using only unique combinations of feature set and class as data) gives an overall accuracy of 89.7% for MBLP and 86.6% for AML with roughly the same distribution of accuracies over the different suffixes. In the remainder of this paper, we will work with the results for the dataset *with* duplications of lexical representations.

For 92.5% of the words, both systems agree on the outcome, and assuming the outcome in the CELEX database to be correct, for 90% of the words they agree on the correct class. Of the 555 cases in which both algorithms predict the same, wrong, class, the majority is due to words with plural suffix $-s$, being assigned to $-e$ or $-(e)n$. E.g. *Autocar, Bar, Jeep, Sheriff* ($-e$ instead of $-s$); *Backhand, Fondue, Tape* ($-(e)n$ instead of $-s$). But many other confusions occur as well. See Tables 4 and 5 for a complete overview. In these Tables, confusion between the different outcomes (classes) is represented. U means Umlaut. E.g. Uer is the class of nouns with plural in $-er$ and with Umlaut, er is the class of nouns with plural in $-er$ without Umlaut.

If we compare the confusion matrices of both systems, we see that they are almost indistinguishable in the confusions made. The Spearman correlation coefficient is 0.999 when taking into account all cells (correct predictions as well as errors). When limited to errors, the correlation is still 0.83 suggesting that both systems make the same confusions. Nevertheless, some of the error categories indicate more divergence: for the cases of grammatical conversion (no suffix is added; $-$ and U in the confusion matrices), the errors made by both algorithms differ more markedly, both the confusion made when assigning

Table 3: Accuracy of AML vs. MBLP on the complete data set using leave-one-out.

Suffix	MBLP Accuracy (%)	AML Accuracy (%)	Frequency
-	96.5	96.1	4651
no Umlaut	96.5	96.1	4402
Umlaut	96.8	96.4	249
-e	92.5	87.0	6656
no Umlaut	92.1	88.2	4646
Umlaut	93.3	84.3	2010
-er	92.7	81.5	974
no Umlaut	92.7	79.4	287
Umlaut	92.7	82.4	687
en	98.3	97.7	11920
s	66.9	46.7	967
Total	95.0	92.0	25168

an incorrect class to these cases (Pearson correlation 0.64), and the type of cases to which conversion is incorrectly assigned (Pearson correlation 0.41). AML especially seems to mistake words much more often for a $-$ or U case than IB1-IG, especially words which should have received an $-e$ plural.

For example, *Almosenier* generates an AML analogical set with the distribution (Uer:0, en:6, Ue:71, -:2880, U:0, er:0, e:834, s:72) whereas IB1-IG finds 3 neighbors at distance 0.3, all with the correct suffix $-e$ (*Harpunier, Pionier, Kanonier*; all with masculine gender and ending in $-ier$). Clearly, looking at local neighborhood only, in combination with assigning more weight to the rhyme of the last syllable and the gender, provides the right sub-generalization here for MBLP.

For all other confusions, correlation is near to or much higher than 0.90, indicating very similar language behavior of both algorithms, except that AML makes significantly more errors than IB1-IG in absolute terms.

Table 4: Confusion matrix for AML. Indicates how many times an exemplar of type as indicated in the rows, was classified as type as indicated in the columns. Correct predictions are on the diagonal.

	-	U	Ue	Uer	e	en	er	s	
-	4191	46	9	3	44	48	5	56	4402
U	78	171	0	0	0	0	0	0	249
Ue	4	0	1893	0	82	26	0	5	2010
Uer	0	0	7	643	31	5	0	1	687
e	33	0	79	30	4318	118	9	59	4646
en	35	13	32	3	103	11708	0	26	11920
er	1	0	0	2	14	0	270	0	287
s	64	1	12	7	153	74	2	654	967
	4406	231	2032	688	4745	11979	286	801	25168

Moving on to other errors made by the algorithms, we see that there are 499 words where AML is correct and MBLP wrong, and more than twice that many (1190 words)

Table 5: Confusion matrix for MBLP. Indicates how many times an exemplar of type as indicated in the rows, was classified as type as indicated in the columns. Correct predictions are on the diagonal.

	-	U	Ue	Uer	e	en	er	s	
-	4231	11	7	1	54	74	2	22	4402
U	8	240	0	0	0	1	0	0	249
Ue	31	0	1694	2	113	167	0	3	2010
Uer	15	0	22	566	70	12	0	2	687
e	104	14	139	32	4097	214	11	35	4646
en	62	6	29	3	159	11649	1	11	11920
er	11	0	0	0	34	13	228	1	287
s	73	3	30	5	185	218	1	452	967
	4535	274	1921	609	4712	12348	243	526	25168

where the reverse holds. When we look at the clustering of errors in these sets of words, we see that even here there is a positive correlation between the types of confusions AML and IB1-IG make when their counterpart is correct.

We have to conclude that, at least for this problem, we find no evidence that the way the AML algorithm works leads to qualitatively different language behavior compared to that when using the conceptually and computationally simpler IB1-IG algorithm. The former leads to significantly lower accuracy, however, and seems to miss certain sub-regularities in the data.

3.3 Related Research

We are not the first to apply these methods to the German plural problem. In (Ramin Charles Nakisa, 1996; Nakisa, 2000), simulation results on CELEX data are reported for nearest neighbour (comparable to IB1, i.e. no feature relevance weighting), Nosofsky's Generalized Context Model (GCM), and a standard three-layer backprop network. The set-up of the experiment is similar to ours (predicting plural class from phonology) but not comparable because of the different data-preprocessing steps resulting in other sets of examples and classes, a different encoding of the phonology (phonetic features instead of segmental syllable structure, no gender), and because of a different methodology, viz. cross-validation instead of leave-one-out. Results were 70.8% for nearest neighbor, 74.3& for GCM, and 82.7% for backprop.

In (Wulf, 1996), AML is also applied to the German plurals problem. Based on a dataset of 703 frequent words, with exemplars encoding phonology and gender, he was able to anecdotally show gang effects and effects of heterogeneity on selected nouns. No accuracy was reported.

Daelemans, Gillis, and Durieux (1997) compared AML and several variants of MBLP on the task of main stress assignment for Dutch. They found that whereas AML outperform IB1, IB1-IG and other variants outperform AML, and are more insensitive to noise. The only other comparison of AML and MBLP we know of (Eddington, 2000) focused on comparing both, as a possible alternative implementation of a single-route model for past tense morphology, to connectionist models, and reported similar results for both when testing on non-words for the past tense.

4 Discussion

In this Section, we refer back to the list of differences noted in Section 2, and discuss these, armed with our new empirical results.

4.1 The Effect of Non-neighbors

In AML, even non-neighbors can in principle affect the decision of the algorithm, as we have seen. MBLP on the other hand relies on local extrapolation: a small neighborhood (typically the nearest neighbor only) is used to extrapolate from. We see that for the German plural at least, the MBLP strategy seems fruitful (e.g. in discovering the subregularity that the plural suffix of masculine nouns in $-ier$ is $-e$). There are 32 cases like that with only two exceptions: *Sire* /zi:r/, plural *Sires* and *Partikulier*, also with plural $-s$. Of these 32, 28 were classified correctly by IB1-IG (the four errors being *Wesir* and *Kurier* which were pluralized $-s$, and *Sire* and *Partikulier*, classified $+e$). On the other hand AML makes these errors as well, and on top of that 6 other errors, including "clear" cases of the subregularity, such as *Almosenier, Fakir, Kanonier, Kurier* as well as *Kashmir, Mudir*.

The problem that the AML algorithm tries to solve by looking at the complete dataset and by classifying subsets of the data as homogeneous and heterogeneous, and that IB1-IG tries to solve by estimating the information gain of each feature, is the problem of *representation relevance*. Which features are most relevant for solving the task? IB1-IG reorganizes the exemplar space (and therefore the distances in it leading to extrapolation of outcomes) by feature weighting. In principle, it is possible to extend the IB1-IG algorithm such that it takes into account all exemplars in memory, by setting the value of k to the number of exemplars, and using the inverse of their distance to the input item to weigh their importance in computing the outcome, but this leads in practice seldomly to better accuracy.

This reliance of IB1-IG on similarity-space reorganization by means of feature weighting, makes the approach of course potentially vulnerable to bad relevance assignments for some features. For example, a known problem with information gain is that it computes the relevance of a feature without taking into account the other features, ignoring possible feature interactions. However, for this problem (and many other linguistic problems we have investigated), it is an accurate and robust heuristic method.

Figure 4.1 shows the relevance assignment of a few different feature weighting methods on our dataset. Gain ratio is a normalized version of information gain (boosting the relevance of features with few values), the χ^2 method uses statistical significance testing to compare the observed distributions of values over classes with their expected distribution (Daelemans, Van den Bosch, and Zavrel, 1999). Interestingly, while the relevance assignment is roughly similar, there are some marked differences, e.g., GAIN RATIO puts more weight on the gender feature, and estimates the relevance of the segmental information lower than the other two methods.

The effect when using these methods in a $k-nn$ algorithm with $k = 1$ on our data (using leave-one-out methodology) is summarized in Table 6. The differences are not important, showing that MBLP is fairly robust to the details of the algorithm for this problem. As could be expected, the algorithm using no reorganization of the exemplar space at all (IB1) performs significantly worse than any of the weighted methods, but it is surprising to see that it outperforms AML. This indicates that all pre-selected features are indeed relevant to solving the task, and that the role of the feature weighting method is in fine-tuning the organization of the exemplar space rather than in re-organizing it. The Table also lists the baseline accuracy when always selecting the most frequent suffix ($-(e)n$), and when probabilistically guessing the outcome (knowing only the distribution of the different classes), called BASELINE1 and BASELINE2, respectively.

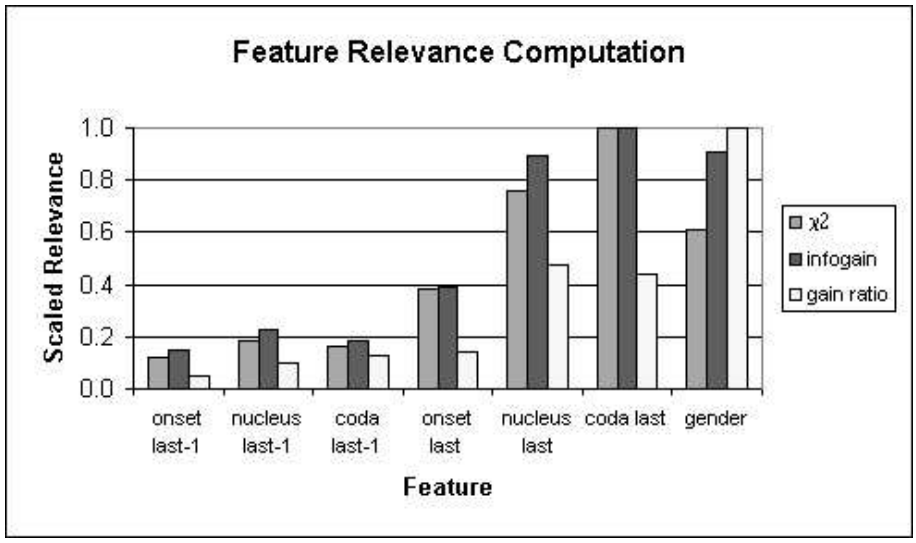

Figure 4: *Feature weights using different weighting methods.*

Table 6: Effect of feature relevance assignment method on accuracy in IB1.

Weighting Method	Accuracy
BASELINE1	46.3
BASELINE2	32.1
IB1	92.6
IB1-IG	95.0
IB1-χ^2	95.1
IB1-GR	94.9

4.2 Value Relevance Weighting

Another potential problem for IB1-IG is the frequency-weighted averaging of the information gain of the different values of a feature to compute the information gain of the feature, as noted earlier. This is a source of robustness (estimation is on the complete dataset), but may at the same time lead to unwarranted underestimation of the relevance of some feature values for some inputs, snowed under in the averaging. Because of the way the algorithm works (treating each value as distinct), AML can assign more or less importance to particular values relative to the particular input it is classifying. In (Skousen, 2000), an example from Finnish past tense is worked out in detail, and it is indeed the case that IB1-IG incorrectly handles this item. However, this is a representation problem more than an algorithmic problem. If a particular value has a high relevance for some types of inputs, it should be assigned a separate feature. It is even possible to explode all values of all features into separate binary features, and use general feature relevance weighting methods on this new representation. This way, the particular Finnish past tense problem can also be solved by IB1-IG (Van den Bosch, personal communication).

Furthermore, whereas it will probably be possible to find similar cases also for the German plural, there will be plenty (60% more) errors made by AML which IB1-IG does not make. In the comparison of the linguistic adequacy of algorithms, the overall accuracy levels are probably more important than casuistic studies. This is of course not the case

for psycholinguistic models; here the algorithms and feature relevance metrics should be compared with human performance and acquisition (see e.g. Eddington (2000)), and overall accuracy is no longer the main evaluation criterion.

4.3 Feature Weighting as Training

Yet another criticism in (Skousen, 2000) of IB1-IG is that because of the feature weighting method used, a training period is needed which makes the approach more akin to connectionism than (to) AML. The important distinction here is that whereas connectionist learning methods such as backpropagation of errors are *batch-learning* methods (cycling several times through all training items until an equilibrium or desired error rate is reached), computing information gain is an *incremental* process, and converges very quickly. For example, Figure4.3 illustrates the convergence of the information gain weights in the differently sized training sets we used to compute the learnability results discussed earlier. Already after a few hundred training items the information gain values are stable, and already from the very first training set, the relative ordering of the relevance of the different features remains basically the same, only the absolute values vary. In addition, the algorithm is very robust to small variations in the specific values of the information gain weights.

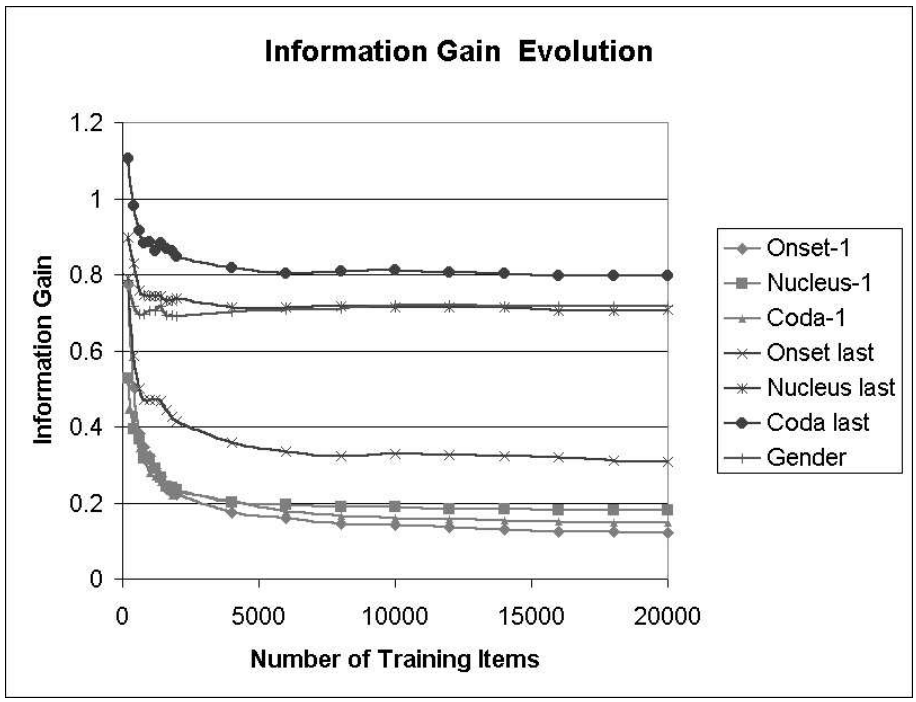

Figure 5: *Convergence of information gain feature weights.*

4.4 (Im)perfect Memory

In language processing tasks, low-frequency events are pervasive. Due to borrowing, historical change, and the complexity of language, most data sets representing language processing tasks contain few regularities, and many subregularities and exceptions. These exceptions and subregularities only concern a limited number of cases, yet, in their small 'pocket of exceptions' in exemplar space, they are productive in that they may correctly

predict the outcome for a previously unseen member of their region. It is impossible for inductive algorithms to reliably distinguish real noise from these pockets of exceptions, so non-abstracting algorithms like IB1-IG should be at an advantage compared to eager learning methods such as decision tree learning or rule induction: 'forgetting exceptions is harmful'. In (Daelemans, Van den Bosch, and Zavrel, 1999), empirical results are provided, and theoretical analysis, supporting this hypothesis.

On the other hand, being based on a minimization of disagreements among data-items, AML is the most powerful statistical test possible, and can be made equivalent to standard statistical procedures by introducing imperfect memory (i.e., introducing a chance that a particular training item is forgotten). Interestingly, and surprisingly from the point of view of the "forgetting is harmful" hypothesis, forgetting 25% and 50% of the training data for the German plural problem does *not* decrease generalization accuracy, which remains at 92%. However, as this is significantly lower than the generalization accuracy of IB1-IG it is unclear what this means. One explanation could be that the way the AML algorithm works on this problem is a form of noise-reduction or smoothing in which the productive subregularities and pockets of exceptions are lost against the more powerful effect of the general tendencies in the dataset (remember that all data items may influence the final decision, not only the local context). The anecdotal evidence about masculine nouns in $-ier$ seems to support this view, but more analysis is necessary. Forgetting part of the data may counter this hypothesised overregularization tendency of AML.

4.5 Computational Complexity and Representational Generality

In (Daelemans, Gillis, and Durieux, 1997), it was argued that an important advantage of MBLP as opposed to the AML algorithm is the fact that the former is linear in the number of features and exemplars, whereas the latter is exponential in the number of features. Massive parallelism does not effectively eliminate this exponential explosion. In (Skousen, 2000), it is argued that the information gain feature relevance weighting in IB1-IG must take into account all psosible combinations of feature values, hence there is no escaping from exponential explosion. But this is clearly not the case. Computation of information gain is linear in the number of data items on which it is computed (all that is necessary is a simple computation on a feature-value outcome-class contingency matrix which can be incrementally collected as experience enters the system). Information gain *does* make the (mostly incorrect) assumption that the features are independent; it is a heuristic. Yet, the empirical tests show that it is an effective and robust relevance estimator for linguistic problems.

Furthermore, the more general approach to similarity used in MBLP allows for the easy and natural definition of similarity for features with numeric and set values, as opposed to AML where only symbolic (nominal) and binary features can be used. Although most language processing representations can be described adequately using nominal features, some linguistic information (e.g., distances between and lengths of linguistic objects like words and utterances; sets of words, phonemes, or letters; etc.) can be more naturally represented using numeric and set valued features.

5 Conclusion

AML and MBLP are similar in spirit, but propose completely different operationalizations of similarity- or analogy-based language processing on the basis of exemplars. In an earlier comparison between AML and MBLP (Daelemans, Gillis, and Durieux, 1997) on the task of main stress assignment in Dutch words, we concluded that for natural language learning tasks there was no clear motivation to use the complex and computationally costly (and with many features computationally intractable) AML algorithm instead of

the more general and less complex class of MBLP algorithms. In this paper we added more substance to this position by analyzing the behavior of AML and IB1-IG on the task of German plural prediction. We found that IB1-IG, a simple MBLP algorithm, significantly outperforms AML, and seems to be better at representing the subgeneralizations of the task. On the other hand, both systems are highly correlated in the errors they make (i.e., the confusions between outcomes they predict), and have very similar learning behavior. Taken together with the additional expressive power and flexibility MBLP offers in handling different types of representations, we stand by our earlier conclusion.

However, additional research is needed to get more insight into the differences between both algorithms in terms of psycholinguistic and linguistic relevance. Work by Gert Durieux, (e.g. Durieux, Daelemans, and Gillis (1998)) suggests that AML is better at learning regularities in the Dutch stress prediction data, whereas MBLP is better at putting to use the predictive power of (small) subregularities.

6 References

Aamodt, A. and E. Plaza. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7:39–59.

Aha, D. W., editor. 1997. *Lazy learning*. Dordrecht: Kluwer Academic Publishers.

Aha, D. W., D. Kibler, and M. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

Atkeson, C., A. Moore, and S. Schaal. 1997. Locally weighted learning. *Artificial Intelligence Review*, 11(1–5):11–73.

Bybee, Joan. 1988. Morphology as lexical organization. In M. Hammond and M. Noonan, editors, *Theoretical morphology. Approaches in modern linguistics*. Academic Press, San Diego, pages 119–141.

Cardie, Claire. 1996. Automatic feature set selection for case-based learning of linguistic knowledge. In *Proc. of Conference on Empirical Methods in NLP*. University of Pennsylvania.

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

Clahsen, Haral. 1999. Lexical entries and rules of language: A multidisciplinary study of german inflection. *Behavioral and Brain Sciences*, 22:991–1060.

Cost, S. and S. Salzberg. 1993. A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10:57–78.

Cover, T. M. and P. E. Hart. 1967. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27.

Daelemans, W., S. Gillis, and G. Durieux. 1994. The acquisition of stress: a data-oriented approach. *Computational Linguistics*, 20(3):421–451.

Daelemans, W., A. Van den Bosch, and A. Weijters. 1997. IGTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423.

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

Daelemans, W., J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 1998. TiMBL: Tilburg Memory Based Learner, version 1.0, reference manual. Technical Report ILK-9803, ILK, Tilburg University.

Daelemans, Walter, editor. 1999. *Memory-based Language Processing.* Special Issue of Journal of Experimental and Theoretical AI, volume 11, number 3, edited byWalter Daelemans. Taylor & Francis.

Daelemans, Walter and Antal van den Bosch. 1992. Generalisation performance of back-propagation learning on a syllabification task. In M. F. J. Drossaers and A. Nijholt, editors, *Proc. of TWLT3: Connectionism and Natural Language Processing*, pages 27–37, Enschede. Twente University.

Daelemans, Walter, Steven Gillis, and Gert Durieux. 1997. Skousen's analogical modeling algorithm: A comparison with lazy learning. In D. Jones and H. Somers, editors, *New Methods in Language Processing.* UCL Press, London, pages 3–15.

Dasarathy, B. V. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* Los Alamitos, CA: IEEE Computer Society Press.

Derwing, B. L. and R. Skousen. 1989. Real time morphology: Symbolic rules or analogical networks. *Berkeley Linguistic Society*, 15:48–62.

Durieux, Gert, Walter Daelemans, and Steven Gillis. 1998. Paper read at the corsendonk round table.

Eddington, David. 2000. Analogy and the dual-route model of morphology. *Lingua*, 110:281–298.

Fix, E. and J. L. Hodges. 1951. Disciminatory analysis—nonparametric discrimination; consistency properties. Technical Report Project 21-49-004, Report No. 4, USAF School of Aviation Medicine.

Jones, D. 1996. *Analogical natural language processing.* London, UK: UCL Press.

Kolodner, J. 1993. *Case-based reasoning.* San Mateo, CA: Morgan Kaufmann.

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

Marcus, G.F., U. Brinkmann, H. Clahsen, R. Wiese, and S. Pinker. 1995. German inflection: The exception that proves the rule. *Cognitive Psychology*, 29:189–256.

Nagao, M. 1984. A framework of a mechanical translation between japanese and english by analogy principle. In A. Elithorn and R. Banerji, editors, *Artificial and human intelligence.* North-Holland, Amsterdam, pages 173–180.

Nakisa, Ramin Charles. 2000. A cross-linguistic comparison of single and dual-route models of inflectional morphology. In Jaap Murre Peter Broeder, editor, *Cognitive Models of Language Acquisition.* Cambridge University Press, pages xxx–yyy.

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

Ramin Charles Nakisa, Ulrike Hahn. 1996. Where defaults don't help: the case of the german plural system. In Garrison W. Cottrell, editor, *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, pages 177–182.

Riesbeck, C. and R. Schank. 1989. *Inside Case-Based Reasoning.* Northvale, NJ: Erlbaum.

Scha, Remko, Rens Bod, and Khalil Sima'an. 1999. A memory-based model of syntactic analysis: data-oriented parsing. *Journal of Experimental and Theoretical Artificial Intelligence*, 11:409–440.

Skousen, R. 1989. *Analogical modeling of language.* Dordrecht: Kluwer Academic Publishers.

Skousen, R. 1992. *Analogy and Structure.* Dordrecht: Kluwer Academic Publishers.

Skousen, Royal. 2000. Analogical modeling. In *Handbook of Quantitative Linguistics*. xxx.

Stanfill, C. and D. Waltz. 1986. Toward memory-based reasoning. *Communications of the* ACM, 29(12):1213–1228, December.

Steven Gillis, Gert Durieux, Walter Daelemans. 2000. Lazy learning: A comparison of natural and machine learning of stress. In Jaap Murre Peter Broeder, editor, *Cognitive Models of Language Acquisition*, pages 76–99. Cambridge University Press.

Wulf, Douglas. 1996. An analogical approach to plural formation in german. In *Proceedings of the Twelfth Northwest Linguistics Conference. Working Papers in Linguistics*, volume 14, pages 239–254.