Ellison & Scobbie (eds), Computational Phonology, EWPCS 8, 157-178, 1993

Learnability and Markedness in Data-Driven Acquisition of Stress

Walter Daelemans Steven Gillis Gert Durieux Antal van den Bosch*

ITK Research Report No. 43, May 1993

Abstract

This paper investigates the computational grounding of learning theories developed within a metrical phonology approach to stress assignment. In current research, the Principles and Parameters approach to learning stress is pervasive. We point out some inherent problems associated with this approach in learning the stress system of a particular language by setting parameters (the case of Dutch), which is shown to be an inherently noisy problem. The paper focuses on two aspects of this problem: we empirically examine the effect of input encodings on learnability, and we investigate the possibility of a data-oriented approach as an alternative to the principles and parameters approach. We show that data-oriented similarity-based machine learning techniques like Backpropagation Learning, Instance-Based Learning and Analogical Modeling working on phonemic input encodings (i) are able to learn metrical phonology abstractions based on concepts like syllable weight, (ii) that their performance can be related to various degrees of markedness of metrical phenomena, and (iii) that in addition, they are able to extract generalizations which cannot be expressed within the metrical framework without recourse to lexical marking. We also provide a quantitative comparison of the performance of the three algorithms investigated.

1 INTRODUCTION

Recently, there has been an increased attention in Computational Linguistics in data-oriented methods taken from Machine Learning or statistical pattern recognition for deriving linguistic knowledge from primary linguistic data. These techniques will help in alleviating the linguistic knowledge acquisition bottleneck and may also provide insight into the way people acquire a language system. At the same time, *computational phonology* has appeared as a mature sub-discipline of the field. In this paper, we try to link these two new areas by showing that machine learning algorithms can be applied to acquire parts of a phonological system.

*Affiliation Walter Daelemans and Antal van den Bosch: Institute for Language Technology and AI (ITK), Tilburg University, P.O. Box 90153, NL-5000 LE Tilburg, Email: walter@kub.nl, antalb@kub.nl. Affiliation Steven Gillis and Gert Durieux: National Fund for Scientific Research (Belgium), University of Antwerp, Universiteitsplein 1, B-2610 Wilrijk, Email: gillis@reks.uia.ac.be, durieux@reks.uia.ac.be. This paper integrates research results to be published in the Proceedings of the European Conference on Machine Learning; Workshop on Machine Learning Techniques for Text Analysis, Vienna 1993, and the Proceedings of the 15th Annual Meeting of the Cognitive Science Society, Boulder Colorado, 1993.

1.1 METRICAL PHENOMENA AND THEORY

Machine learning of metrical phenomena is an interesting domain for exploring the potential of particular machine learning techniques, and more generally, to study the role Machine Learning can play in theory formation (the *computational grounding* of a learning theory).

First of all, the assignment of stress in polysyllabic monomorphemic words, the subject of this paper, has been fairly well studied in metrical phonology. Within this framework, the stress patterns of numerous languages have been described in considerable detail. Thus, a solid theoretical framework as well as elaborate descriptions of the linguistic data are available.

Secondly, the domain of metrical phenomena can be studied as a (relatively) independent problem domain (unlike problems in other linguistic domains such as, for instance, linguistic pragmatics, that typically have multiple dependencies with other domains like syntactic and/or semantic phenomena).

Thirdly, metrical phenomena exhibit a number of interesting characteristics that makes them well-suited for testing the capacity of machine learning algorithms to generalize, as well as their ability to handle irregularities. On the one hand, stress assignment appears to be governed by a number of solid generalizations. For the purpose of this study, a lexicon of Dutch polysyllabic monomorphematic words was compiled (the lexicon will be described in more detail in section 1.3). We found that approximately 80% of the 4868 monomorphemes are regular according to a state-of-the-art metrical analysis (Trommelen & Zonneveld 1989, 1990). The remaining 20% have to be dealt with in terms of idiosyncratic markings (specification of a lexical foot, exceptions to extrametricality, a combination of these two exception mechanisms, or simply a marking of the irregular pattern in the lexicon). On the other hand, the domain exhibits a large number of local ambiguities, or, in other words, it can be said to be noisy. For instance, taking the aforementioned lexicon, a metrical encoding in terms of syllable weight[1] was performed. This revealed that only 44 of the 89 possible weight strings were unambiguous with respect to stress assignment. This ambiguity can be exemplified as follows: if we take a string of three LIGHT syllables, the three possibilities of the Dutch stress system occur:

VV-VV-VV
*pa*nama (Panama): antepenultimate stress
pi*ja*ma (pyjamas): penultimate stress
para*plu* (umbrella): final stress

In short, it can readily be seen that the microcosm of metrical phonology is endowed with generalizations as well as irregularities. This is a phenomenon characteristic of the macrocosm of the linguistic system in general.

1.2 MACHINE LEARNING OF METRICAL PHENOMENA

Recently, computational learning models that specifically address the problem of how to learn the regularities of stress assignment have been proposed: Gupta & Touretzky (1991), Dresher & Kaye (1990), Nyberg (1991). They all approach the learning problem from the angle of the 'principles and parameters' framework (Chomsky 1981). In this approach the learner comes to the task of language learning equipped with a priori knowledge incorporated in a universal grammar that constrains him to entertain only useful generalizations. More specifically, the a priori knowledge consists of a finite set of parameters, the values of which have to be fixed

[1] Metrical analyses of Dutch assume four levels of syllable weight: super light (schwa) light (VV) heavy (V+C) super heavy (VV+C, (V)V+CC). The last category leaves room for further differentiation.

by the learner so as to arrive at the grammar of his 'local' language. Starting from a finite set of parameters, each with a finite set of values, the number of possible grammars developed by the learner is restricted to a finite set. It is assumed that universal grammar specifies a number of parameters relevant to the metrical domain (see Dresher & Kaye 1990). The computational models add a learning theory to the linguistic notion of universal grammar. This learning theory specifies what aspects of the data are relevant to each parameter, and it also determines how the data processed by the learner are to be used to set the values of the parameters. Common to the systems referred to is that they try to fix the values of parameters relevant to the metrical domain. The specific approaches taken differ, however, with respect to important dimensions. Dresher & Kaye (1990) and Nyberg (1991) explicitly incorporate a set of parameters, while Gupta & Touretzky (1991) aim at discovering them. Dresher & Kaye implement a deterministic learner while Nyberg's is a non-deterministic one.

The research reported in this paper aims at exploring the potential of learning algorithms that share a DATA-DRIVEN (empiricist) mode of learning instead of the nativist approach exemplified by the research described in this section. We investigate how far we can get in acquiring noise-tolerant generalizations without presupposing a lot of a priori knowledge (although even in data-driven algorithms, knowledge is also present to a greater or lesser degree in the data encodings used). A second goal of the present research is a qualitative comparison of the results of our data-driven approach to the constructs and insights of the metrical framework.

Dresher & Kaye (1990, Dresher, 1992) explicitly mention TABLE-LOOKUP (storing weight strings with their associated stress string) as an uninteresting data-driven approach, mainly because in their view it is empirically inadequate as it cannot generalize to new cases. Yet, we show that a learning theory based on analogical reasoning can be associated with such an approach. Our experiments also show that an underlying (syllable weight) representation need not be derived explicitly from surface representations (phonemic representation). This transformation is a problem which is faced, but not solved by the approaches mentioned. Church (1992) in a reaction to Dresher (1992) also mentions (data-driven) table-lookup as an alternative, but glosses over the problems of noise (ambiguous patterns) and of how to arrive at a syllable weight representation. We agree with Church however that even if the data-oriented (table-lookup) approach *overgenerates* in the sense that impossible stress systems could be learned, this is not something *the learner* should be concerned with.

1.3 STRESS ASSIGNMENT IN DUTCH

In order to introduce the qualitative analysis of the results of our experiment, a short presentation of some basic facts about the stress system of Dutch appears to be appropriate. The most straightforward way to present stress assignment in Dutch is by reviewing the settings of the relevant metrical parameters (see Dresher & Kaye 1990, Trommelen & Zonneveld 1989, 1990):

P1	The word-tree is strong on the [Left/Right]	Right
P2	Feet are [Binary/Unbounded]	Binary
P3	Feet are built from the [Left/Right]	Right
P4	Feet are strong on the [Left/Right]	Left
P5	Feet are quantity sensitive [Yes/No]	Yes
P6	Feet are quantity sensitive to the [Rhyme/Nucleus]	Rhyme
P7	A strong branch of a foot must itself branch [No/Yes]	Yes
P8A	There is an extrametrical syllable [No/Yes]	Yes
P8	It is extrametrical to the [Left/Right]	Right

This configuration of parameters yields a metrical tree such as the one depicted in Figure 1. Binary feet, labeled s (strong) and w (weak) are built from right to left on top of syllable rhymes. Feet are quantity-sensitive to the rhyme and closed syllables may not occupy the w-position of a binary foot. A word tree which is strong to the right is built on top of the feet and main stress can be determined by folowing the path containing exclusively s-labels.

As indicated in Figure 1, the rightmost syllable is extrametrical. In Dutch, extrametricality is restricted to VX-rhymes, where X stands for V or C. This condition results in the extrametricality of VV- and VC-rhymes, but is not stretched further to include super heavy syllables (which can only occur in word final position).

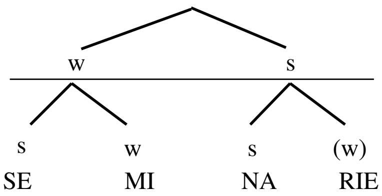

Figure 1: Metrical tree for **seminarie**

Extrametricality needs two further qualifications. First of all, Dutch is fairly idiosyncratic in the sense that extrametricality applies after foot formation and before word tree formation, a phenomenon called 'late'-extrametricality. In this way, extrametricality does not affect regular foot formation but it does influence directly the construction of the word tree. Secondly, extrametricality is subject to percolation from 'head-position' (the s-position of a foot or a monosyllabic foot). This ensures for instance that monosyllabic feet remain invisible for the word tree formation rule (after word tree formation they are incorporated into the word tree by STRAY ADJUNCTION). At the same time, the final syllable in Figure 1 is a weak syllable, and hence, since extrametricality is not allowed to percolate upwards from within the weak, non-head position, the final foot is still incorporated in the word tree.

This setting of the metrical parameters defines a window of three syllables from the right word edge. Generally speaking, stress on the penultimate syllable is the normal case, stress on the antepenultimate syllable is made possible by the extrametricality of the final syllable, except if the latter is super heavy since in that case the final syllable is stressed.

Deviations from this pattern are handled as follows:

- **Lexical Feet [F]**. The mechanism of idiosyncratically assigning a lexical foot stipulates that a syllable marked with the feature [F] behaves as an exception to regular foot formation in that it constitutes a monosyllabic foot. In this way, the final monosyllabic foot of words with a final light syllable (as in '*Pa*nama') becomes extrametrical and stress can land on the antepenultimate syllable of the word.[2]

- **Exceptions to the extrametricality rule [-ex]**. This mechanism indicates that words marked as [-ex] are to be withdrawn from the regular application of the extrametricality rule. The aim is to attract stress to that final syllable that would be extrametrical in the regular case. For instance, several VC- final words, such as kolo*nel colonel*, receive final stress.

- **Lexical feet in conjunction with exceptions to extrametricality [F], [-ex]**. The third mechanism combines the two preceding ones: it marks the final syllable so that it is assigned a monosyllabic foot, and subsequently this syllable is withdrawn from the regular application of the extrametricality rule by a [-ex] marking, resulting in final stress. Cases in point are words with a final open syllable that nevertheless receive final stress such as para*plu umbrella*.

The three exception mechanisms have in common that the relevant words have to receive a marking in the lexicon. This also holds for those words that can still not be satisfactorily treated by the mechanisms discussed. For instance, there are exceptions to the general rule that words with a final super heavy syllable have final stress, such as *al*taar *altar*.

In order to estimate the generality of the metrical categories, a lexicon of Dutch polysyllabic monomorphematic words was compiled. Our data consisted of 4868 polysyllabic monomorphemes. The lexicon was extracted from the CELEX lexical database.[3] Only words that could be unambiguously characterized as monomorphemes were selected for our data set. Proper nouns were withdrawn from the dataset. Our lexicon constitutes a representative sample of the monomorphemes of the language.

In the metrical analysis, five cases can be distinguished (between brackets their frequency in the lexicon): (i) the regular (R), unmarked case (80.44%); (ii) a mechanism that intrudes into foot formation: [F] (3.86%); (iii) a mechanism that affects word-tree formation: [-ex] (7.15%); (iv) a combination of (ii) and (iii) (5.38%); (v) the irregular cases (I) (3.16%). These five possibilities can be scaled according to their markedness: the regular case (i) is of course the least, the irregular case (v) the most marked. In-between these extremes, possibilities (ii) and (iii) are less marked than (iv): the latter requires two features while for the former only one feature is sufficient.

This scaling can also be performed on a more fine-grained level. From foregoing short presentation of the basic facts of Dutch stress assignment, it appears that words with a light or heavy final and prefinal syllable are the problematic cases. The way they are analysed in a metrical framework is summarized in Table 1.

When we take the first type in Table 1, it appears that Penultimate stress is the unmarked case, Antepenultimate stress requires one feature, and Final stress is the most marked case since two features are needed.

[2] Trommelen and Zonneveld (1990) propose that in VX-VV-VC words with penultimate stress (such as 'me*ce*nas') the penultimate VV-syllable has a lexical foot, which constitutes a departure from the regular case in which the final syllable is treated as such.

[3] CELEX contains 103,778 lemmas and 399,816 wordforms. It was compiled on the basis of the INL corpus of present-day Dutch; more than 42 million words in a variety of text types.

Table 1: Stress patterns in Dutch words with light and heavy syllables

Type	Stress Pattern		
	Final Stress	Penultimate Stress	Antepenultimate Stress
VV-VV-VV	[-ex][F]	R	[F]
VX-VC-VV	[-ex][F]	R	I
VX-VC-VC	[-ex]	R	I
VX-VV-VC	[-ex]	[F]	R

1.4 THE PARAMETER SETTING PROBLEM

A parametric approach that aims at universal validity will eventually have to deal with the irregular, exceptional, and language specific details of the linguistic system. At present this appears to be a problem. For instance, Dresher & Kaye (1990) explicitly require that the input be completely transparant. They dedicate a specialized module to determining if there exist obvious conflicts (such as the ones illustrated above for Dutch). Eventually, a brute force learner is invoked to deal with similar input.

They also indicate that the set of parameters will undoubtedly have to be extended (see also Gupta & Touretzky 1991). But keeping the present set of parameters as sufficient, for the sake of the argument, a number of serious problems turn up when we try to analyze how a learner of Dutch might fix the values of the parameters. Two examples may suffice to illustrate the point. Parameter 6 relates to quantity sensitivity, and more specifically determines if a language is quantity sensitive to the rhyme or to the nucleus. If the former is the case, closed syllables and long nuclei behave similarly with respect to stress, while in the latter case only branching nuclei are heavy. It is not clear how these cues for Parameter P6 can be used in Dutch where closed syllables do indeed behave as open syllables with long vowels but this is only so for heavy closed syllables and not for super heavy ones.

Another problem arises with respect to the extrametricality parameters 8A and 8. Dresher & Kaye (1990: 189) point out that extrametricality is a difficult problem since the cue "(...) presence of stress at the left or right edge of a word is enough, in this system of parameters, to rule out extrametricality at the edge." But lack of stressed peripheral syllables is not a sufficient condition for extrametricality. In the case of Dutch there is a firm number of words exhibiting final stress (in our lexicon of 4868 polysyllabic monomorphemes: 39.59%). Thus how can the learner determine that for Dutch a parameter setting amounting to right extrametricality is appropriate given a huge number of words with final stress? Moreover the theory should provide a way to disentangle the cues for setting parameter 8A (extrametricality) and parameter 6 (quantity sensitivity) since a branching rhyme is subject to extrametricality except for super heavy syllables (with either a branching nucleus or a branching coda). If such a construction can be found it would account for 68.35% of the cases with final stress. For the remaining words with final stress the theory should find a way to discover the application of exception mechanisms, viz. [-ex] (18.06% of words with final stress), and [F][-ex] (13.60% of words with final stress).

Here the theory presents two escape mechanisms: ETCHING and a THEORY OF EXCEPTIONS. Etching basically provides a mechanism for detecting and filtering out noise from the input stream by making decisions about parameter settings sensitive to frequency. The problem with this approach is that there is no principled way to determine the saturation level: what amount of data should be encountered before a parameter is set to its marked value? For instance, if 39.59% of all monomorphemic words have final stress, would the saturation level be reached for setting the extrametricality parameter (erroneously) to its marked value? The second proposal involves a more principled account of exceptions, viz. a deterministic learner is able to handle exceptions if they invariably go in the unmarked direction since exceptions can never be taken as evidence for setting a parameter to its marked value. In the case of Dutch this is not unproblematic: as a language with right extrametricality, final stress is exceptional. Thus, if the unmarked setting of the extrametricality parameter is [8A YES] (assuming extrametricality as the unmarked value) the huge amount of words with final stress do not confirm the unmarked but the marked option. Hence, exceptions do not point into the unmarked direction but in the marked direction, and it remains uncertain if the learner will ever reach the correct parameter setting.

In the light of the above problems, we investigated the learnability of Dutch stress assignment in a machine learning experiment. Our data set consisted of 4868 monomorphemic Dutch words. First we investigated whether the performance of some of the proposed data-oriented machine learning algorithms differed on the stress assignment learning task (experiment 1), then we tried to relate the learning performance of two of the algorithms to the sort of encoding used and to linguistic theory formation within the metrical framework (experiment 2). Before we describe the experimental results, however, we still have to introduce the learning algorithms used.

1.5 THE LEARNING ALGORITHMS

We have experimented with three different learning algorithms: Backpropagation of errors in feedforward networks (BP), Analogical Modeling (ANA) and Instance-Based Learning (IBL). All three algorithms are supervised (a number of training items is provided). IBL is an incremental algorithm, BP and ANA are batch learning algorithms. BP is sub-symbolic (it uses microfeatures extracted from input activation patterns), IBL and ANA are symbolic[4].

In the three learning algorithms, similarity plays a central role: similar instances have similar categories. Both IBL and ANA make explicit use of similarity-based reasoning. They use a similarity metric to compare items, and use the items most similar to a test item as a basis for making a decision about the category of the latter. BP too, uses similarity (or analogy), but more implicitly. Again, an input pattern activates an output pattern which is similar to the activation pattern of those items that are similar to the new item. Complexity is added by the fact that an intermediate layer of units "redefines" similarity by extracting features from the activation patterns of the input layer. We will see that in our version of IBL, an information-theoretic metric is used to achieve a similar result.

1.5.1 Backpropagation Of Errors

We chose Backpropagation of errors (BP) (Rumelhart, Hinton & Williams, 1986) as one of our inductive techniques because it has empirically been shown to be relatively successful in learning natural language processing sub-problems (e.g., Daelemans & Van den Bosch, 1992,

[4] A more detailed classification of these algorithms in the space of possible machine learning algorithms is given in Gillis et al. 1992.

present a 3-layer BP network which learns to hyphenate Dutch words with relatively high accuracy; Weijters & Hoppenbrouwers, 1990, present a similar network which learns to map Dutch text to phonemic speech). The main reason for experimenting with BP for learning relatively complex problems like hyphenation or stress assignment, is that BP networks are alleged to be able to extract generalisations and sub-generalisations from their training data, as well as store exceptions to these generalisations. However, there are limitations to BP network capabilities. BP learning is not guaranteed to converge to optimal performance (i.e. it can end up in local minima). A consequence of this is that although a multi-layered network may be able in principle to represent the solution to any mapping problem, this property is not of much help because the designer of such a network is confronted with a large search space of variable network parameters (e.g., size of the hidden layer, learning rate, number of training cycles) which may affect learning and performance of the network considerably, but which cannot be determined by rule. Experimenters can therefore almost never be sure that their results are optimal.

In all BP simulations, we implemented a 3-layer network. All simulations were run on PlaNet v5.6, a connectionist network simulator written by Yoshiro Miyata. The input layer contained 25 units, the hidden layer 37 units, and the output layer 3 units. All simulations were run 50 epochs. Before training, all connection weights were initialized with random floating point values between -.5 and .5. We used a learning rate of 0.2 and a momentum of 0.9.

1.5.2 Instance-Based Learning

Instance-based learning (IBL, Aha et al. 1991) is a framework and methodology for incremental supervised machine learning. The distinguishing feature of IBL is the fact that no explicit abstractions are constructed on the basis of the training examples during the training phase. A selection of the training items themselves is used to classify new inputs. IBL shares with Memory-Based Reasoning (MBR, Stanfill and Waltz, 1989) and Case-Based Reasoning (CBR, Riesbeck and Schank, 1989) the hypothesis that much of intelligent behaviour is based on the immediate use of stored episodes of earlier experience rather than on the use of explicitly constructed abstractions extracted from this experience (e.g. in the form of rules or decision trees). In the present context of learning linguistic mappings, the hypothesis would be that much of language behaviour is based on this type of memory-based processing rather than on rule-based processing. In linguistics, a similar emphasis on analogy to stored examples instead of explicit but inaccessible rules, is present in the work of, amongst others, Derwing and Skousen (1989). IBL is inspired to some extent on psychological research on exemplar-based categorization (as opposed to classical and probabilistic categorization, Smith and Medin, 1981). Finally, as far as algorithms are concerned, IBL finds its inspiration in statistical pattern recognition, especially the rich research tradition on the nearest-neighbour decision rule (see e.g. Devijver and Kittler, 1982, for an overview).

The operation of the basic algorithm is quite simple: for each pattern to be assigned a category (test item), it is checked whether this pattern has been encountered in the training set earlier. If this is the case, the category of the training item is assigned to the new item (or the category most often associated with the training item in case of ambiguous patterns). If the test item has not yet been encountered, its similarity to all items kept in memory is computed, and a category is assigned based on the category of the most similar item(s). The performance of an IBL classifier crucially depends on the selection of training items to be kept in memory, and the similarity metric used. In these experiments, we "remembered" all training items, and only experimented with the similarity metric.

When using a Euclidean distance metric (geometrical distance between two patterns in pattern space), all features are interpreted as being equally important. But this is of course not necessarily the case. We extended the basic IBL algorithm proposed by Aha et al. (1991) with a technique for assigning a different importance to different features. Our approach to the problem of weighing the relative importance of features is based on the concept of Information Gain (IG, also used in learning inductive decision trees, Quinlan, 1986), and first introduced (as far as we know) in IBL in (Daelemans and Van den Bosch, 1992) in the context of a syllable segmentation task. The idea is to interpret the training set as an information source capable of generating a number of messages (the different categories) 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. It is computed by the following formula where p_i (probability of category i) is estimated by its relative frequency in the training set.

$$H(D) = -\sum_i p_i log_2 p_i \tag{1}$$

For each feature (position in the patterns), it is now computed what the *information gain* is of knowing its value. To do this we have to compute the average information entropy for this feature and subtract it from the information entropy of the database. To compute the average information entropy for a feature, we take the average information entropy of the database restricted to each possible value for the feature. The expression $D_{[f=v]}$ refers to those patterns in the database that have value v for feature f, V is the set of possible values for feature f.

$$H(D_{[f]}) = \sum_{v_i \in V} H(D_{[f=v_i]}) \frac{|D_{[f=v_i]}|}{|D|} \tag{2}$$

Information gain is then obtained by equation three, and scaled to be used as a weight for the feature during similarity matching.

$$G(f) = H(D) - H(D_{[f]}) \tag{3}$$

While the approach taken makes the IBL algorithm not completely incremental (in our experiments, the IG value is computed on the basis of the complete training set), we have experimented with an incremental version (updating the IG values with every new item), with the same results after less than a hundred training patterns. The second experiment was performed with an incremental version of the technique.

Figure 2 shows the information gain values for each attribute in the encoding; for all categories taken together (IG), and for each category independently (Information Gain of ultimate syllable IG(L); of penultimate syllable IG(BL); and antepenultimate syllable IG(BBL)). Horizontally, the different attributes are represented (number of syllables #S; syllable weight ultimate syllable W(L); nucleus N(L), and presence of onset O(L); and syllable weight penultimate syllable W(BL), nucleus N(BL), and presence of onset O(BL)). The data encoding will be more fully explained in the next section.

Unlike information gain for all attributes taken together, information gain of each category independently was not used further in our experiments because it did not improve overall generalization accuracy. It did improve accuracy for category BBL (antepenultimate), however. This simple information-theoretic metric already provides some insight into the problem: nucleus identity and syllable weight of the last syllable are clearly the most important features overall, and should be assigned most weight in similarity matching. The metric also shows that the encoding used does not offer many clues for the prediction of the antepenultimate stress. Word length (in number of syllables) provides most information gain here. The (rounded) information gain value is used to weigh similarity matching.

Information gain values for each attribute

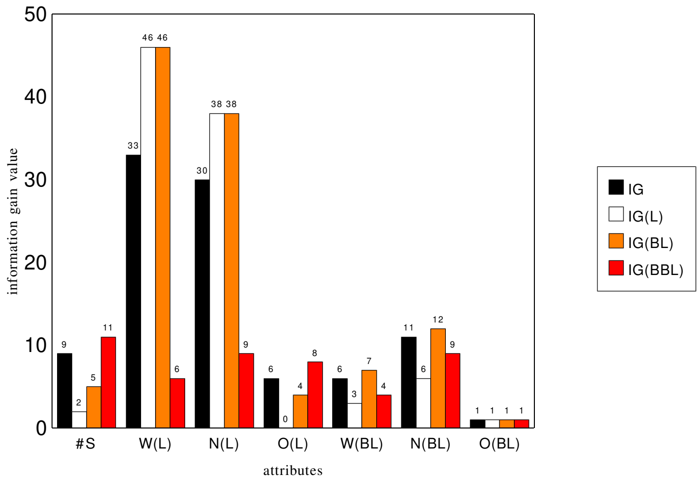

Figure 2: Information Gain values for each attribute

1.5.3 Analogical Modeling

Analogical Modeling (Skousen 1989) is another similarity-based framework, meant to provide an alternative to rule-based linguistic descriptions as a model of actual language usage. The main assumption underlying this approach is that many aspects of speaker performance are better accounted for in terms of 'analogy', i.e. the identification of similarities or differences with other forms in the lexicon, than by referring to explicit but inaccessible rules (see Derwing & Skousen 1989 for an overview of psycholinguistic research supporting this assumption). The notion of 'analogy' is given an operational definition in terms of a matching process between an input pattern and a database of stored exemplars. The result of this matching process is a collection of examples called the analogical set, and classification of the input pattern is achieved through selection from this set.

ANA thus shares some important characteristics with IBL: the main source of knowledge in both approaches is a database of stored exemplars. These exemplars themselves are used to classify new items, without intermediate abstractions in the form of rules. In order to achieve this, an exhaustive database search is needed, and during this search, less relevant examples need to be discarded. The main difference between both approaches lies in the way this selection is made. In our information-gain extension of IBL, different weights are attached to each feature in a pattern, so that correspondences between informative features are favoured over similarities between less informative features. In ANA essentially the same effect is achieved without precomputing the relative importance of individual features. Instead, all features are equally important initially, and serve to partition the database into several disjoint classes of exemplars. Filtering out irrelevant exemplars is done by considering properties of these classes 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 some more detail.

The first stage in the matching process is the construction of subcontexts; subcontexts are just classes of exemplars, 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 the input pattern 325, which represents a syllable weight encoding of a word like 'astronaut' as an example, eight different subcontexts would be constructed, 325, $\overline{3}25$,$3\overline{2}5$, $32\overline{5}$, $\overline{32}5$, $\overline{3}2\overline{5}$, $3\overline{25}$ and $\overline{325}$, where the overstrike denotes complementation. Thus, exemplars in the class 325 share all their features with the input pattern, whereas for those in $\overline{32}5$ 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 generalizing 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 exemplars 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
3	2	5	325
3	2	-	325 $32\overline{5}$
3	-	5	325 $3\overline{2}5$
-	2	5	325 $\overline{3}25$
3	-	-	325 $3\overline{2}5$ $32\overline{5}$ $3\overline{25}$
-	2	-	325 $\overline{3}25$ $32\overline{5}$ $\overline{3}2\overline{5}$
-	-	5	325 $\overline{3}25$ $3\overline{2}5$ $\overline{32}5$
-	-	-	325 $\overline{3}25$ $3\overline{2}5$ $32\overline{5}$ $\overline{32}5$ $\overline{3}2\overline{5}$ $3\overline{25}$ $\overline{325}$

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.

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 weighed according to its distribution across different supracontexts. 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. For example, input patterns with a very strong cue, such as /ə/ in the final syllable, give rise to analogical sets which display a lot of variation in the other features, whereas for minor subregularities, such as words ending in /ium/, only exemplars with slight deviations from the input pattern will survive in the analogical set. To finally categorize the input pattern, either the predominant category in the analogical set or the category of a randomly chosen member of this set is chosen. In our experiments, we adopted the former approach.

Analogical Modeling has been applied to a number of different problem domains, where the degree of regularity ranges from entirely categorical to fairly idiosyncratic. In all of these cases, the model appears to perform well and captures the relevant generalizations. Moreover, an analogical approach allows for smoother transitions between boundary cases and is able to deal with missing or redundant information.

2 EXPERIMENT 1

In a first experiment, we quantitatively compared the performance of the three data-driven learning algorithms on learning the stress assignment task, and studied the learning curve produced by each algorithm by presenting different training set sizes.

2.1 METHOD

To be relatively certain that the performance results reported approximate the *true* error rate, we set up a 10-fold cross-validation experiment (10-fold CV, Weiss & Kulikowski, 1991). In this set-up, the dataset is partitioned ten times, each time with a different 10% of the dataset as the test set, and the remaining 90% as training set. For each of the ten simulations, we also varied the size of the training set from 500 items to the full training set size, with increments of 500. This adds up to 90 different training set - test set divisions for each algorithm tested.

2.2 DATA CODING

As far as the representation of words is concerned, for this experiment, we chose a representation combining abstract elements from the metrical framework (such as syllable weight) with more concrete properties of the word: its length in number of syllables, the presence or absence of a syllable onset in the final and prefinal syllable, and the nucleus (vowel) of the final and prefinal syllable. For instance, the word **nirvana** was encoded as *32a+2a+* (3 syllables, both final and prefinal syllable have an onset indicated by the plus sign, are light syllables, and have phoneme *a* as their nucleus).

Word	Number of Syllables 3		
Syllable	onset present	nucleus	weight
Penultimate	+	a	2
Ultimate	+	a	2

In BP, the values of input attributes were encoded as randomly generated bit strings (one input unit for each bit), using the smallest possible number of bits. For instance, for the attribute representing the identity of the nucleus of the last syllable, 6 units are needed. In total, each input pattern was represented using a 25-unit string. The output category is encoded locally, i.e., each outcome is represented by one unit in the output layer, resulting in a 3-unit output encoding.

In both ANA and IBL, training and test patterns are straightforwardly encoded as strings of attribute values.

2.3 RESULTS

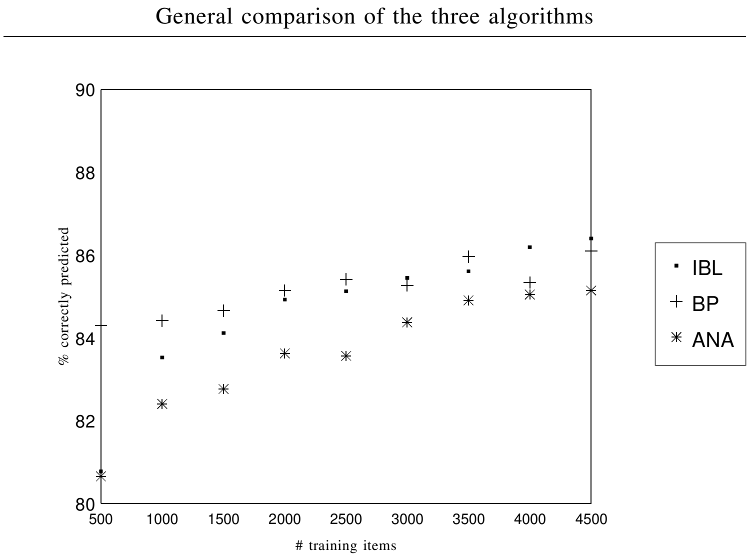

Figure 3: General comparison of the three algorithms

In Figure 3 the global performance of the different algorithms is plotted. The mean success rate in the 10-fold cross-validation experiments was calculated for each training set (500 items, 1000 items, ... 4500 items)[5]. The general picture can be summarised as follows[6].

- With few training items, BP reaches the best performance of the three algorithms, IBL and ANA are far less successful when only a limited number of training items is available. More specifically, BP scores best when trained with 500 items, i.e., it obtains

[5] Note that the logarithmic fits have very high r^2 values: BP: $r^2 = .818$, IBL: $r^2 = .963$, ANA: $r^2 = .979$.

[6] For details about the results of all experiments we performed and their statistical analysis, we refer to Gillis et al. 1992.

the best success score with few training items (BP: 84.29% correctly classified, IBL: 80.77%, ANA: 80.65%).

- IBL eventually reaches the highest peak performance. ANA is the least successful of the three algorithms. In absolute terms, IBL has the highest score: trained with 4500 items, IBL reaches a success rate of 86.40%. This success score is better than for ANA (85.14%) and BP (86.10%).

If we pool the results of all the 10-fold CV experiments per algorithm, we find that IBL and BP do not differ significantly, but IBL and BP differ significantly from ANA. When we compare the results at the level of the individual training set sizes, however, it appears that the performance of the three algorithms does not differ significantly. In other words, when we compare the results per training set (i.e., compare ANA, IBL, and BP results for a training set of 500 items, 1000 items, and so on), none of the comparisons turn out to be statistically significant.

When we study the global learning curves more closely, it is striking that BP scores high at the beginning but does not improve significantly beyond that point. ANA and IBL on the other hand significantly improve their scores. In both cases the result for 500 training items differs significantly from the score obtained from the experiments with more training items.

Comparison of success rates

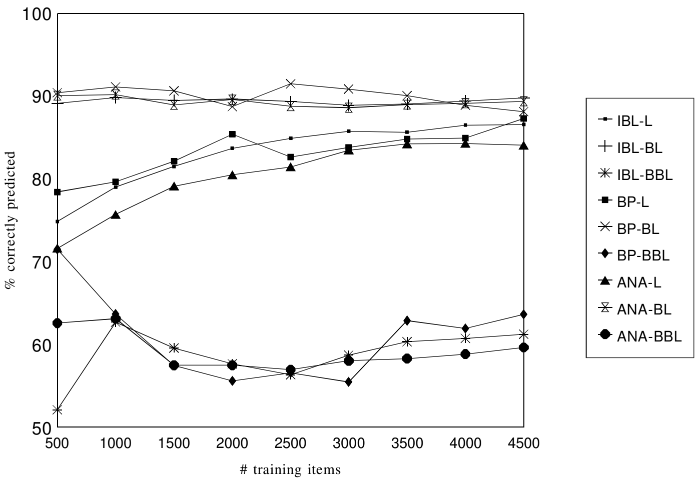

Figure 4: Comparison of success rates relative to target category

When we go down to the level of the individual target categories, i.e., stress on the final, penultimate and antepenultimate syllable, the three algorithms appear to yield highly comparable results. In Figure 4 the success rates of the algorithms are specified per target category. The graph shows three clusters of lines. On top we find the results for 'stress on the penultimate syllable' for the three algorithms. Somewhat lower, the results for 'stress on the final syllable' cluster together. At the bottom we find 'stress on the antepenultimate syllable'. Thus, the three algorithms agree in that stress on the penultimate syllable can be best predicted, followed by stress on the final syllable. Stress on the antepenultimate syllable appears to be very difficult to predict. But again, the results of the three algorithms show essentially the same picture.

In Figure 4 it can readily be seen that the learning curves are fairly comparable. The curves representing stress on the penultimate syllable do not show any major fluctuation. The three algorithms score around 90% accuracy and that rate remains the same irrespective of the size of the training set. For stress on the antepenultimate syllable, there are some fluctuations, but no significant ones. Only in the case of stress on the final syllable there is a significant learning effect: the three algorithms improve their success rates: trained with 500 or 1000 items they score significantly less well than when trained with more items. Beyond that training set size, there is still an increase of the success rate in absolute figures, but that increase is not significant.[7]

The conclusion that can be drawn from these findings is that the three algorithms behave similarly with respect to the three target categories and the general profile of the learning curves for those categories. However, the similarity is more profound than mere overall appearances. An investigation of the differences between the results specified per target category reveals no significance.

There is one more aspect of the comparison of the three algorithms that deserves attention. When we study the learning curves in Figure 4, the curves for 'stress on the final syllable' and 'stress on the prefinal syllable' are relatively far apart initially, but due to a learning effect 'stress on the final syllable' is eventually almost as accurately predicted as 'stress on the prefinal syllable'. It was already noted that there is indeed a significant learning effect for 'stress on the last syllable'. That effect is exhibited by the three algorithms. Moreover, an analysis further shows that initially the prediction for 'stress on the prefinal syllable' is significantly better than those for 'stress on the final syllable' but eventually a comparison of the two result categories does not show significance any more. Again the three algorithms behave essentially in the same way in this respect. There is one difference however. When we investigate where the precise point can be located where the highly significant difference between the predictions for the two last syllables turns into a non significant one, the three algorithms do not agree completely: for IBL that point is reached with a training set of 2500 items (i.e., with 2000 training items the success rate for stress on the final syllable are still significantly lower than the success rate for stress on the prefinal syllable), for ANA it is 3000 items and for BP turns out to be 3500 items.

Taken together, these results point very strongly in the direction of a very high similarity of the performance of the three algorithms. There is one perspective from which the three algorithms can be seen to perform differently and that is when we look at the global picture of all the results. In that case there is one algorithm, viz. analogical modeling, that performs not as well as the other two algorithms. But once we start investigating the performance of the algorithms at a more detailed level, the differences remain but they cannot be shown to be statistically significant. In other words, the three algorithms' success scores are essentially the same. The close similarity of the results between BP and IBL on the one hand versus ANA on the other hand, led us to the elimination of BP as a learning algorithm in the second experiment, in which we were more interested in qualitatively different behaviour than in performance comparisons.

[7] All training sets approximate the distribution of categories in the dataset: 7% antepenultimate, 40% final, and 53% penultimate.

3 EXPERIMENT 2

The first experiment was mainly concerned with the quantitative aspects of the performance of the learning algorithms, and with a comparison of their performance. In a second experiment the aim was to conduct a qualitative analysis of the algorithms' learning abilities. More specifically we explored the role of the encoding used (syllable weight versus phonemic encoding) on learnability, the types of errors made by the algorithms, and how those errors relate to a metrical analysis. The learners also faced the problems associated with non-transparent input, i.e., as was the case in the first experiment, equal input patterns can have different target categories leaving the learner with the task of resolving these local ambiguities in some way or another.

3.1 METHOD

In this experiment, the leaving-one-out method was used. For this purpose, each item in our dataset in turn is selected as the test item, with the remainder of the dataset as training set. We therefore get as many simulations as there are items in the dataset. This computationally very costly method has as its major advantage that it provides the best possible estimate of the true error rate of a learning algorithm (Weiss & Kulikowski 1991).

3.2 DATA CODING

The data were encoded (i) as strings of syllable weights of the last three syllables of the word (encoding-1), and (ii) using the phonemic information contained in the rhyme projections of the last three syllables (encoding-2). For instance, the word **nirvana** was encoded as follows:

	Encoding-1	Encoding-2	
Syllable	weight	nucleus	coda
Antepenultimate	3	I	r
Penultimate	2	a	-
Ultimate	2	a	-

3.3 RESULTS

3.3.1 Analysis of General Performance

Both algorithms attain an overall success rate of around 81% for encoding-1 and around 87% for encoding-2. Specified to the level of individual target categories, it appears that both algorithms agree that stress on the penultimate syllable can be more efficiently predicted than stress on the final syllable. Stress on the antepenultimate syllable is fairly difficult to predict.

	Encoding-1		Encoding-2	
	IBL	ANA	IBL	ANA
Total	81.2	81.2	87.6	86.9
Final	70.5	70.5	86.0	85.2
Penultimate	93.7	93.7	91.9	92.2
Antepenultimate	49.6	49.3	64.5	57.6

A comparison of the results for the two encodings yields the global result that both algorithms take advantage of the details provided in encoding-2: in general an encoding in terms of weight strings does not lead to better results than an encoding in which the nucleus and the coda are fully specified. The difference between the results for the two encodings are statistically significant ($p < .0001$).

Specified at the level of the individual target categories, the results are more diversified: the encoding in terms of weight strings (encoding-1) yields better results (for the two algorithms) for the unmarked case, viz. stress on the penultimate syllable. For the two other target categories, the second encoding scheme yields superior (and statistically significantly better) results. In other words, weight strings are sufficient to capture the most general and metrically unmarked case, which is the most frequently occuring one, quite well. However, that information is insufficient to extract satisfactorily the regularities involving the more marked cases, viz. final and antepenultimate stress. Hence, an important inprovement in the systems' performance can be seen for the latter two categories. In other terms, it appears that the regularities governing stress on the final and stress on the antepenultimate syllables require information present in encoding-2 and absent in encoding-1. This implies that in abstracting syllables weights from the rhyme projections, essential information is lost.

Thus the question turns up which generalizations within the domain are captured by training the systems with the two encodings?

3.3.2 Weight Strings versus Rhyme Projections

The fact that a number of general characteristics of stress assignment can be captured given the weight string encoding can be shown by scrutinizing strong generalizations within the domain that can be formulated in terms of syllable weight: (i) we already indicated that super heavy final syllables are not eligible for final extrametricality, consequently they receive stress almost without exception (in our lexicon: 1015 words with final super heavy syllable receive final stress, while in only 56 words stress is not final); (ii) super light syllables can never be stressed, moreover super light final syllables are almost without exception preceded by a stressed syllable (1316 words in our lexicon have prefinal stress when the final syllable is super light and only 11 have stress on the antepenultimate).

Final	ANA		IBL	
Syllable	encod-1	encod-2	encod-1	encod-2
super heavy				
-VVC	1015	978	1015	1008
	(94.77)	(91.32)	(94.77)	(94.12)
-VCC	302	303	302	301
	(82.51)	(82.79)	(82.51)	(82.24)
total	1317	1281	1317	1309
	(91.65)	(89.14)	(91.65)	(91.09)
super light	1316	1319	1316	1317
	(99.17)	(99.40)	(99.17)	(99.25)

It can readily be inferred from this table that the generalizations that can be formulated in terms of syllable weight are captured by both systems. The phonemic encoding (encoding-2) does not yield highly superior results; on the contrary, with respect to super heavy syllables in word- final position, slightly worse results are found (None of the differences are significant at the 5% level or below).

Turning to the categories specifically relevant for extrametricality, it appears that less stringent generalizations are discovered by the algorithms. Light and heavy final syllables are considered to be extrametrical in the current account, thus a prefinal stress pattern is to be expected. This expectation is only realistic, however, if only extrametricality is playing in determining stress assignment. That is not the case: for -VV final words only 65.15% actually receive penultimate stress, and for -VC final words only 43.69%. These figures sharply contrast with those for the super heavy and the super light final syllables, for which the algorithms found satisfactory generalizations. The results in the next Table show that similar generalizations were out of reach for the light and the heavy final syllables.

Final	ANA		IBL	
Syllable	encod-1	encod-2	encod-1	encod-2
light -VV	774	911	773	928
	(67.60)	(79.56)	(67.51)	(81.05)
heavy -VC	545	717	548	708
	(56.83)	(74.77)	(57.14)	(73.83)

In comparison with the success scores for the super heavy and super light final syllables, the success scores of the light and heavy ones are inferior. Even for the phonemic encoding, a success rate of 80% can hardly be reached. It can be noted that the success scores for the encoding in weight strings are below the phonemic encoding for both algorithms (Differences are significant at $p < .0001$).

3.3.3 Exception Mechanisms and Markedness

As was already indicated in the description of stress assignment of Dutch monomorphemic words, several exception mechanisms have been invoked to account for the apparent lexical diffusion that appears to govern the words with final light and heavy syllables. When we classify the results of our experiments according to those categories a highly illuminating picture occurs (Figure 5). In Figure 5 the results for IBL are given, the results for ANA (Figure 6) are highly similar.

1. The regular cases are almost perfectly predicted when the algorithms are trained with a weight string encoding (IBL- 1: 99.59%, and ANA-1: 99.57%). This encoding is not able to deal with the lexically marked words: the success scores for the four exception codings hardly reach 10%. For both algorithms, the results for the regular category are significantly ($p < .01$) better for the weight string encoding than for the phonemic encoding of the rhyme.

2. The exceptional cases reach a fairly acceptable level of accuracy when the algorithms are trained with a phonemic encoding of the rhymes of the three last syllables. All differences between encoding-1 and encoding-2 reach significance at the 1% level or below.

3. A comparison of the learning results for the phonemic encoding with the 'markedness'-scale presented within the metrical framework (section 1.3), immediately reveals that

Success rates per 'metrical category'

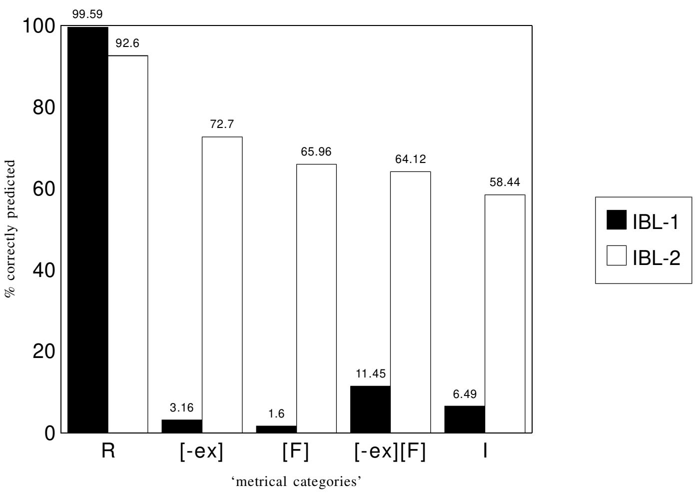

Figure 5: IBL Performance relative to Metrical Category

there is a remarkable correspondence between the two: the more marked a category from a metrical point of view, the lower the success rate of that category in the learning experiments. Hence, the regular cases are fairly well learned by both algorithms, and they both show poor performance for the irregular cases. With respect to the exception mechanisms in-between these two extremes, marking of an exception with respect to extrametricality ([-ex]) and the marking of a monosyllabic lexical foot ([F]) have better scores than the category that combines the two features. Thus, the markedness relations between these exception mechanisms are reflected in lower success scores.

These results lead us to the conclusion that there is a close correspondence between markedness in terms of exception mechanisms invoked for particular classes of words and the learnability of those words: for unmarked classes of words the learning algorithms reach a superior success score in comparison to the more marked classes.

Does this close correspondence between markedness in the metrical framework and learnability in the computational context also hold when we flesh out the results for specific types of words? In Table 2 the information from Table 1 is repeated and the learning results for IBL and ANA are added.

At first sight, relative markedness from a metrical point of view appears to be a good predictor of the success scores of the learning algorithms. Take the VV-VV-VV words as an instance. The regularly stressed type (stress on the penultimate syllable) has, by far, the best success score. A somewhat lower score is obtained for the words with antepenultimate stress. These words are more marked than the regulars: they need a single exception feature. Final stress is obtained for words with two features; this category, the most marked of the three, has the worst score. Thus metrical markedness is reflected in the success scores of both algorithms. A similar finding holds for the VX-VC-VV words: R > [-ex][F] > I.

The relationship between markedness and success scores does not seem to be as strong when we consider the two bottom rows of Table 2. For VX-VC-VC words, the irregular antepenultimate stress, the most marked category, is very poorly predicted by both algorithms.

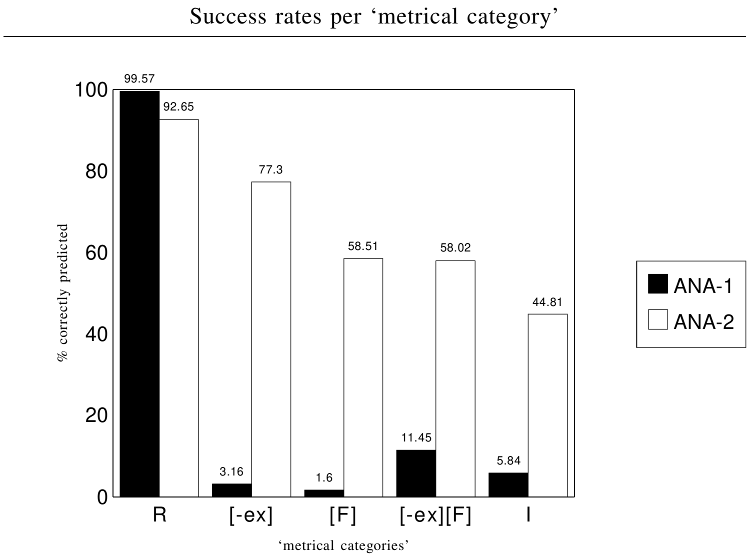

Figure 6: ANA Performance relative to Metrical Category

But the regular penultimate stress is hardly better predicted by ANA than the more marked final stress, IBL predicts the more marked category even better than the unmarked regular one. In the bottom row a similar situation is found: the regular case is less well predicted than the more marked ones for IBL, and less well predicted by ANA than the more marked final stress words and only slightly better predicted than the more marked penultimate stress words. Although these results appear to contradict the relationship between markedness in metrical terms and success scores of the algorithms, this contradiction can be explained by a closer analysis of the learning patterns.

When we scrutinize the results for the individual patterns of words, both IBL and ANA appear to have discovered subregularities in the data that are not (even, cannot be) accounted for in the metrical framework. Indeed, in the latter, syllables are used in the analysis as far as their weights are concerned. The identity of individual vowels and consonants is not taken into account in the constructions of metrical trees. And hence, for the word types considered, important subclasses of words that behave homogeneously cannot be identified. But given the phonemic encoding used in the learning experiment, the algorithms quite successfully traced these subregularities. For instance, the high success scores for final stress in words with a VX-VV-VC pattern is partly due to the fact that almost half of these words (48%) have /ɛ/ in their final syllable. They are successfully stressed by ANA (100%) and IBL (93.51%). Both algorithms seem to have discovered the more general subregularity in the lexicon with respect to these words, viz. words ending in /ɛ/ almost unanimously prefer final stress (94.48% final stress, 5.22% penultimate stress, and 0.3% antepenultimate stress on a total of 326 words). This outspoken homogeneous behaviour of words with /ɛ/ in their final syllable is reflected in the success scores of the algorithms: 88.34% for IBL and 92.03% for ANA. The apparently regular words with final stress were accurately stressed by IBL (95.47%) and ANA (99.30%).

The ability of the algorithms to trace subregularities in the data and the breadth of that ability is further illustrated in the following example: 25% of the VC final words have /ʌ/ in their final syllable. This category of words is an almost perfect example of lexical diffusion:

Table 2: Stress patterns in Dutch words with light and heavy final syllables

	Stress Pattern		
Type	Final Stress	Penultimate Stress	Antepenultimate Stress
VV-VV-VV	[-ex][F]	R	[F]
IBL	60.00	83.45	77.78
ANA	55.71	93.53	66.67
VX-VC-VV	[-ex][F]	R	I
IBL	65.63	91.06	0.00
ANA	46.88	99.19	0.00
VX-VC-VC	[-ex]	R	I
IBL	83.33	80.65	33.33
ANA	91.67	93.55	0.00
VX-VV-VC	[-ex]	[F]	R
IBL	67.16	73.33	65.29
ANA	82.09	65.00	68.82

48.08% have penultimate stress and 44.23% antepenultimate stress. The algorithms appear to have made finer distinctions within this set of words: both IBL and ANA appear to have detected that (Latin) words with /i/ in the prefinal syllable and /ʌm/ in the final syllable act as a fairly homogeneous category with respect to stress (95.24% of these words have antepenultimate stress and 4.76% penultimate stress). Hence the success scores for IBL and ANA are equal for this category: 96.43% of the words are correctly stressed. Moreover, as an 'intermediate step', they appear to have found that words with /i/ in the prefinal syllable and /ʌ/ in the final syllable have an outspoken preference for antepenultimate stress: ANA predicts stress correctly in 93.86% of the cases and IBL in 86.84%. When we look at these words in our lexicon, we find that 14.04% have prefinal stress, of which 11 are bisyllabic. The stress pattern of the bisyllabic words is predicted correctly by the algorithms. Two words receive final stress and both algorithms err. 84.21% of those words have antepenultimate stress - ANA correctly predicts that stress pattern, but IBL misses eight words. The eight words appear to have a final /ʌs/ syllable, which was recognised by ANA but not by IBL.

These results lead us to the conclusion that the correspondence between markedness in the metrical framework and ease of learning by the algorithms also holds at the level of individual types of words. Although at this level the correspondence is not accross the board, apparent exceptions can be accounted for by fact that the algorithms traced subregularities in the data that cannot be captured using the less fine-grained weight strings used in the metrical framework.

4 CONCLUSION

A metrical analysis reveals that Dutch simplex words can adhere to a regular pattern laid out by the default rules of stress assignment. In specific instances, however, exceptional markings of lexical items are invoked to arrive at correct stress assignment. The analysis eventually requires full lexical marking for irregular patterns. As such we arrive at a markedness scale: the regular patterns occupy the unmarked position, the irregular patterns the most marked position, and in-between the other possibilities can be ordered according to the number of exception features. The learning experiments with ANA and IBL show that the more categories of words are metrically marked, the less accurately they are learned. Hence, a correspondence was found between markedness and ease of learning by the artificial learning algorithms. This correspondence also holds for individual categories of words that consist of the same syllabic weight string, but that nevertheless exhibit different stress patterns. It was found that the more marked stress patterns (in terms of exception features) are less accurately learned. Hence, metrical markedness and ease of learning also correspond on the level of individual categories of words. Our experiment has thus provided *computational grounding* (Gupta & Touretzky, 1991) to the application of the metrical phonology framework to Dutch stress assignment.

As regards the problem of how stress can be acquired, the second experiment shows that data-oriented learning algorithms incorporating similarity-based reasoning on the basis of phoneme representations, are able, *without* recourse to lexical marking, to correctly assign stress to many cases that are considered marked in metrical phonology. The forms considered regular in metrical phonology are indeed the forms best learned by our algorithms, but there are additional "subregular" classes of forms about which regularities can be extracted on the basis of phonological form. These regularities are used by the algorithms to predict stress positions in unseen, similar, cases. This can be accounted for by the fact that the similarity-based algorithms extract subregularities in the data that cannot be captured using the machinery of metrical phonology. We therefore believe that the results of our experiments show that a data-driven (empiricist) computational learning theory can be considered a serious alternative to learning theories based on a P&P approach.

Finally, our experiment comparing IBL and ANA to the more well-known backpropagation in connectionist networks approach for the stress assignment task showed that IBL and BP have a better generalization performance than ANA on the stress assignment task. Broken down to the level of the individual training sets (experiments), there are no statistically significant differences, however.

Acknowledgements

The research of Steven Gillis and Gert Durieux was supported by a research grant from the National Fund for Scientific Research (Belgium), and a research grant "Fundamentele Menswetenschappen" (8.0034.90). We are especially grateful for the comments of Georges De Schutter and Arthur Dirksen on our work. Previous versions of part of this research have been presented at CLIN 1992 (Tilburg), the IPRA Colloquium (Antwerp), the 1992 Computational Phonology Workshop (Edinburgh) and TIN 1993 (Utrecht). We are grateful to the participants of these meetings for comments and advice.

References

Aha, D., Kibler, D. & Albert, M. 1991. Instance-Based Learning Algorithms. *Machine Learning* 6, 37-66.

Chomsky, N. 1981. Principles and parameters in syntactic theory. In Hornstein, N. & Lighfoot, D. eds. *Explanations in linguistics: The logical problem of language acquisition.*

London: Longman. pp. 32-75.

Church, K. 1992. Comment on Computational Learning Models for Metrical Phonology. In: R. Levine (ed.) Formal Grammar: Theory and Implementation, O.U.P., 1992, 318-326.

Daelemans, W. & van den Bosch, A. 1992. Generalization Performance of Backpropagation Learning on a Syllabification Task. In: M.F.J. Drossaers and A. Nijholt (eds.) *Connectionism And Natural Language Processing*. Proceedings Third Twente Workshop On Language Technology, pp. 27-38.

Derwing, B. L. & Skousen, R. 1989. Real Time Morphology: Symbolic Rules or Analogical Networks. *Berkeley Linguistic Society* 15: 48-62.

Devijver, P.A. & Kittler, J. 1982. *Pattern Recognition. A Statistical Approach*. London: Prentice-Hall.

Dresher, E. 1992. A Learning Model for a Parametric Theory in Phonology. In: R. Levine (ed.) Formal Grammar: Theory and Implementation, O.U.P., 1992, 290-317.

Dresher, E. & Kaye, J. 1990. A computational learning model for metrical phonology. *Cognition* 34: 137-195.

Gillis, S., G. Durieux, W. Daelemans, & A. van den Bosch. 1992. Exploring Artificial Learning Algorithms: Learning to Stress Dutch Simplex Words. *Antwerp Papers in Linguistics* 71.

Gupta, P. & Touretzky, D. 1991. Connectionist models and linguistic theory: Investigations of stress systems in language. Unpublished ms.

Nyberg, E. 1991. A non-deterministic, success-driven model of parameter setting in language acquisition. Unpublished PhD, Carnegie Mellon University.

Quinlan, J. R. 1986. Induction Of Decision Trees. *Machine Learning* 1: 81-106.

Riesbeck, C. K. & Schank, R.S. 1987. *Inside Case-Based Reasoning*. Hillsdale: Erlbaum.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. 1986. Learning Internal Representations By Error Propagation. In D.E. Rumelhart & McClelland, J.L. (Eds.), *Parallel Distributed Processing: Explorations Into The Microstructure Of Cognition*, Vol. 2. Cambridge, MA: MIT Press, pp. 216-271.

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

Smith, E.E. & Medin, D.L. 1981. Categories and Concepts. Cambridge, MA, Harvard University Press.

Stanfill, C. & Waltz, D.L. 1986. Toward Memory-based Reasoning. *Communications of the ACM* 29: 1213-1228.

Trommelen, M. & Zonneveld, W. 1989. *Klemtoon en Metrische Fonologie*. Muiderberg: Coutinho.

Trommelen, M. & Zonneveld, W. 1990. Stress In English And Dutch: A Comparison. Dutch Working Papers in English Language and Linguistics 17.

Weijters, A. & Hoppenbrouwers, G. 1990. Netspraak: Een Neuraal Netwerk Voor Grafeem-Foneem-Omzetting. *Tabu*, 20: 1-25.

Weiss, S. & Kulikowski, C. 1991. *Computer Systems That Learn*. San Mateo: Morgan Kaufmann.