MEMORY-BASED LEARNING MODELS OF INFLECTIONAL MORPHOLOGY: A METHODOLOGICAL CASE STUDY

EMMANUEL KEULEERS* WALTER DAELEMANS**

1. INTRODUCTION

The paper investigates the memory-based learning (MBL) paradigm as a model of productive linguistic behavior in the domain of Dutch noun plural inflection. We first sketch the origin and background of the MBL approach, to then provide a short overview of Dutch noun plural inflection along with a detailed description of the use of MBL models for inflectional morphology. Results of a large number of MBL simulations on three related tasks of noun plural inflection are analyzed in considerable detail. In particular, we discuss the differential effects of varying core parameter configurations of the MBL algorithm, issues of representation of source exemplars, and different definitions of inflection as a classification task. Finally, we consider these results in relation to current practices in the optimization of model parameters and in the analysis and evaluation of simulation results.

2. BACKGROUND

The central claim of the MBL paradigm is that decisions about new facts are based on re-use of stored past experiences. In this approach, learning is storage of exemplars in memory, and processing is analogical reasoning on stored exemplars. The idea has a long history in cognitive science, with a few pioneering insights going back to pre-Chomskyan linguistics (De Saussure 1916, Bloomfield 1933). Related ideas can also be found in current research in both exemplar-based (Skousen 2002) and cognitive linguistics (Croft and Cruse 2003). In psychology, exemplar-based approaches have been proposed to model human categorization behavior (e.g. Estes 1994). An algorithmic operationalization of the approach was developed in the statistical pattern recognition literature from the 1950s onwards (Fix and Hodges 1951) with the "near-

* Center for Psycholinguistics, University of Antwerp.
** Department of Linguistics, CNTS, University of Antwerp.
This research was supported by grant G.0519.04N of the Flemish Fund for Scientific Research (FWO).

est neighbor rule" modeling generalization as either extrapolation from one nearest neighbor (1-NN) or from more than one (*k*-NN). The algorithm found considerable favor in artificial intelligence where it was rubricated under the headings of "case-based reasoning", "memory-based reasoning", "instance-based learning" etc. (see Daelemans and van den Bosch 2005, for an overview and for the application of MBL in computational linguistics).

In modeling inflectional morphology, a memory-based approach assumes morphological generation to be a function of either lexical retrieval or similarity-based reasoning on lexical representations of word forms, where computation of similarity is defined on the basis of phonological, orthographical, or even semantic representation features.

At least three components are necessary to describe an MBL model: a knowledge base containing exemplars (also called "instances", "examples" or "experiences") with an associated class; a function that describes how similar two exemplars are; and a decision function that determines the class of a new exemplar as a function of the classes associated with its *k* nearest neighbors. Exemplars can be thought of as bundles of feature values and the similarity between exemplars as a function of the similarity between values. The simplest possible such model is the 1-NN model, where the class of the most similar exemplar determines the target class. Originally, nearest-neighbor algorithms were defined only for numerical features, but today MBL has been extended to encompass a wide variety of methods for assessing similarity for both numerical and nominal features. In this paper, our simulations make use of the TiMBL 6.0 system (Daelemans *et al.* 2007), a software package that collects a number of variants of MBL.

It is important to bear in mind that the goals of implementing an MBL-model are not the same in different domains. In most computational linguistics tasks, the goal is to maximize performance accuracy, that is, to be able to classify both new and existing exemplars correctly. In computational psycholinguistics, the goal is to characterize human generalization behavior, that is, to classify new exemplars the way humans do. We will examine the implications of this distinction in more detail later on.

In describing MBL as a model of inflectional morphology, three points are worth emphasizing. First, MBL takes the view that each inflected form is valuable. There is no need for developing representations that abstract away from experience. Second, word inflection is considered to be a fully context-dependent process. Finally, MBL makes a principled distinction between retrieval and generalization.

2.1 Exhaustive Storage

In MBL, all exemplars in a domain are stored on a par, and each classification step is governed by the same similarity and decision functions. Each response

or classification step is the result of an analogical process, consisting in the comparison of the target exemplar to previously stored exemplars and the consequent generalization of the class of the most similar known exemplars to the target. Whereas most cognitive models presuppose the explicit representation of generalizations as abstractions from sets of exemplars, and the explicit storage of irregular exemplars as exceptions to these generalizations, MBL does not make this distinction and keeps all exemplars available to potential extrapolation in analogy-based processing.

2.2 Context Dependence

Because there is no representational difference in MBL between regular and irregular exemplars, it can be seen as a one-route context-dependent model. In that respect, it keeps company with other one-route approaches, such as analogical modeling (Skousen 2002), connectionist pattern associators, the general context model (Nosofsky 1988), or context-dependent rule-based models (Albright and Hayes 2003). It thus contrasts with dual-route models (Pinker 1999, Clahsen 1999), where a context-dependent component is complemented with a default mechanism that is context-independent.

2.3 Generalization Is Not Retrieval

In MBL models, production of known inflected forms is carried out through simple retrieval; the analogical route is resorted to only for the production of inflected forms of unknown exemplars. This contrasts with models that use the same mechanism to produce target forms for known as well as for novel exemplars, e.g. the Rumelhart and McClelland (1986) model of English past tense inflection. The distinction between generalization and retrieval follows from a difference in the learning process. While a connectionist model has a learning phase in which weights are adjusted for most known inflected forms to be correctly produced, MBL models do not have such a learning phase. Because MBL models base the inflection of new forms directly on analogy to stored exemplars, they are also known as *lazy* learning models.

Over the last several years, MBL has been used to model lexical productivity in different domains. A number of studies successfully applied MBL to the modeling of experimental evidence. Hahn and Nakisa (2000) used a simple *k*-NN model to predict plural forms for novel German nouns, Krott, Schreuder and Baayen (2002) and Krott *et al.* (2007) investigated the choice of linking morphemes in novel Dutch and German compounds, Keuleers *et al.* (2007) studied Dutch noun plural inflection, and Eddington (2000) focused on English past tense formation. Substantial work was also devoted to lexical reconstruction tasks in the domains of Dutch word stress (Daelemans, Gillis and Durieux 1994) and German plural formation (Daelemans 2002). In lexi-

cal reconstruction, predictions are not validated against experimental data, but rather against a wide range of attested lexical evidence. Part of the vocabulary data is used as a knowledge base for constructing a model which is eventually tested on the remaining vocabulary used as a test material. Finally, some MBL work was aimed to model child language acquisition data (Gillis, Durieux and Daelemans 2000).

3. MODELING DUTCH NOUN PLURAL INFLECTION

Dutch has two frequent and productive inflectional affixes for plural formation, *-s* and *-en,* the latter of which is phonologically realized as /ə/. The two suffixes are almost, but not completely, in complementary phonological distribution, so that the plural suffix for a Dutch noun is to a relatively high degree predictable given the noun phonology. For instance, *voet* ('foot') – like most other nouns ending in an obstruent – takes the *-en* suffix in its plural *voeten*, and *bakker* ('baker') – like most other nouns ending in a sonorant consonant preceded by /ə/ – takes the *-s* suffix in *bakkers*. Phonological rules like these (De Haas and Trommelen 1993) can account for the plurals of about three quarters of Dutch monomorphemic nouns.[1] While rule-based descriptions of the Dutch noun plural system offer a clear and concise view of the domain, our goal here is to understand Dutch noun plural inflection (and inflectional morphology in general) in a memory-based learning framework.

3.1 Tasks

Each model will be run on three tasks: one lexical reconstruction task, and two pseudo-word plural production tasks. The lexical reconstruction task consists in predicting the plural forms of 5% of the nouns in the lexicon on the basis of all remaining ones. In the pseudo-word tasks, the model is expected to match the plural forms produced by the majority of participants in two controlled experiments. In the first experiment (Baayen *et al.* 2002), subjects produced plurals for a set of 80 pseudo-words with up to four syllables. In the second experiment (Keuleers *et al.* 2007), subjects produced plural forms for 180 mono- and disyllabic pseudo-words.[2] In both experiments, pseudo-words covered a wide range of phonological conditions thought to affect plural formation.

[1] A second factor determining a Dutch noun's plural suffix is the perception of whether a word is a borrowing, in which case the *-s* suffix is often preferred. This factor will not be considered in the present study, but see Keuleers *et al.* (2007) for a memory-based learning approach that takes borrowings into account.

[2] The experiment elicited productions in three *spelling* conditions. Only plural productions for pseudo-words in the *no spelling* and *Dutch spelling* conditions are considered here. Plural productions for pseudo-words in the *English spelling* condition were ignored.

3.2 Memory

In implementing an MBL model, the first step is choosing the exemplars that will make up the stored knowledge base. In the case of inflectional morphology, a corpus-derived lexical database such as CELEX (Baayen, Piepenbrock and Gulikers 1995) is often used as the source for exemplars. While the basic assumption in MBL is that every single item is stored, the set of exemplars stored in the model's knowledge base is in fact subject to several practical limitations. The task being modeled is the most obvious constraining factor. It is assumed that only exemplars for which a relevant target class can be determined are relevant. In noun plural formation, the relevant target class is a label from which it is possible to determine the plural inflection of a noun from its corresponding singular form. In practice, this means that only nouns for which both singular and plural forms are attested are relevant exemplars. CELEX lists 19,351 such nouns.

In building the knowledge base, it is common practice to leave out exemplars that occur below a given frequency threshold, based on the intuition that exemplars that are more frequent are more salient. There are two reasons why this is, in our opinion, unjustified. First, one of the core assumptions of the MBL paradigm is that each exemplar is relevant to generalization behavior. Second, low frequency exemplars play an important role in generalization. For instance, Baayen (2001) demonstrated that the productivity of an inflectional pattern rises with the number of *hapax legomena* showing that pattern. There is considerable evidence that type frequency and not token frequency is a determining factor for generalization (Bybee 1995). This is also supported by the observation that irregular instances (for example, of English verb inflection) often present disproportionally high frequencies. The practice of leaving out exemplars that occur below a particular token frequency is difficult to justify. In our opinion, the proper course of action is to include all exemplars that are expected to be already known in the learning condition being modeled. In the current study, we tried to model the adult learning state, and so we assume that even very low frequency forms were present as stored exemplars in the model's knowledge base.

Another practical reason to limit the number of exemplars in memory is that a large number of stored exemplars may increase the computational cost of a simulation. However, the MBL implementation in TiMBL takes advantage of very efficient data compression. Typically, a TiMBL simulation using one combination of model parameters and a full set of several thousands of exemplars takes a few seconds to run on a standard personal computer. Nonetheless, whenever computation time is a real concern, we suggest reducing the number of exemplars by random selection rather than by frequency.

3.3 Class

In the experimental tasks we aim to model, participants are asked to produce inflected forms for pseudo-words. The traditional approach in MBL is to consider this as a simple classification problem, where the model's task is to predict the relevant inflection class of each input form. In this case the choice is between the productive suffixes *-en* and *-s*. As the plural suffix will be extrapolated from exemplars stored in the model's memory, each exemplar is labeled with the suffix it selects for plural formation. For exemplars that do not form their plural with either suffix, a third class label is used.

The main advantage of this approach is that it is fairly straightforward to compare the model's predictions to experimental results, as participants' responses are categorized using the same labels. However, there are also potential drawbacks. Class labels abstract away from relevant features of actually produced inflections. For instance, phenomena of consonant alternation, which occur for some nouns but not for others, are ignored. The label *-en* is used for both *hand* (plural *handen*) and *kant* (plural *kanten*). However, the final consonant in *hand* is unvoiced in the singular /hant/ and voiced in the plural /handə/, while the final consonant of *kant* is unvoiced both in the singular /kant/ and the plural /kantə/. This does not mean that the labeling of experimental results is erroneous. Since our focus is on whether *-en* or *-s* is used, phonological realization details can be seen as further refinement steps of this analysis. Participants had the freedom to produce alternations, and therefore did not perform the same task as the model. Likewise, the *a priori* partition of productive plural formation processes into two classes, with all other processes being grouped under a single label, may be too much of a simplification. For instance, in Dutch, many nouns borrowed from Latin, Italian, and Greek keep their etymological plural form (e.g., *museum-musea*), and these processes are productive at least to a certain extent. Borrowing terminology from data compression, we may say that the class labeling approach to Dutch plural noun inflection is *lossy*, in the sense that it does not allow us to perfectly recover the plural form from the singular form.

Another relevant observation is that the way in which classes are defined may affect the MBL algorithm quite extensively, hence leading to important differences in the output. We will come back to this point in the sections on feature weighting and decision. Suffice it to emphasize now that, while class labels may be increasingly refined to include processes such as consonant alternation or extended to account for irregular processes, the algorithm needed to assign the correct class labels to each exemplar becomes more and more complex with each such refinement. In turn, this increases the possibility of errors.

A radical alternative to such a class-based conceptualization of the inflection task is the use of a generic mapping algorithm yielding a description of how a form in memory is transformed into a target form. Such a complex

description – essentially a transformation function – can then effectively be used as class label. In this approach there is no need to define the possible class labels beforehand. As a result, the class detection algorithm does not become more complex when more class labels are introduced. Furthermore, the approach has two important properties. First, since the target inflected form can always be recovered from its input form and the corresponding transformation function, we can consider each transformation function as a *lossless* class label. Second, the classification task becomes equivalent to a production task, since the transformation function applied to target forms produces fully specified forms.

For this purpose, we used the Ratcliff/Obershelp pattern recognition algorithm (Ratcliff and Metzener 1988). When applied to a pair of symbol sequences, the algorithm operationalizes the steps through which one sequence can be transformed into the other one. Unlike the Levenshtein distance, the algorithm does not yield the minimal number of editing operations needed, but rather aims at attaining a maximally psychologically plausible string transformation. In van den Bosch and Daelemans (1999), a similar transformation function approach is successfully used in an MBL engineering approach to morphological analysis.

A consequence of using a transformation function is that the number of classes becomes very large. One of the goals of this study is to compare the traditional method of assigning pre-defined class labels based on linguistic categories to the alternative approach of using a transformation function. Comparative results will be assessed over different data sets and different combinations of other parameters.

3.4 Features and Similarity

As the main goal of an MBL model is to extrapolate the class of new exemplars based on their similarity to stored exemplars, here we will discuss different aspects of similarity in some detail. There are four main issues to be addressed in this respect: what information domains define the similarity space where exemplars are compared; which domains are especially relevant to the task being modeled; how can we make exemplars comparable for the relevant information; how is similarity computed on the basis of inter-exemplar comparison.

3.4.1 Choice of Information

In languages such as English and Dutch, the primary factor determining the choice of a particular inflectional marker is phonology.[3] As already noted

[3] In theories such as Pinker (1999) and Clahsen (1999), it is proposed that a default process that does not take into account lexical information explains a large part of inflection. Keuleers *et al.* (2007) have argued that this account is very unlikely for Dutch plural inflection.

above, for about three quarters of Dutch noun types, the plural form can be predicted by applying deterministic rules to the phonological properties of singular forms. In an MBL approach to Dutch plural inflection, we will assume that the inflectional pattern of a non-stored exemplar *e* is best predicted on the basis of the inflectional pattern of the stored exemplars phonologically most similar to *e*. While we will limit ourselves here to phonological information, it is noteworthy that other possible factors could in principle be taken into account. For example, Baayen and Moscoso del Prado Martín (2005) demonstrated that in Dutch, German, and English regularly and irregularly inflected verbs have different semantic densities: the inclusion of semantic information in an MBL model would allow semantic similarity between exemplars to contribute to the prediction of inflectional forms. Moreover, it has been argued that pragmatic similarity between exemplars plays a role in inflection (Keuleers *et al.* 2007).

Linguistic accounts of the Dutch noun plural system reach a very adequate description by focusing on the rhyme of the final syllable and the noun's stress pattern. This means that while Dutch may contain words with more than two syllables, a model in which exemplars are compared only on the basis of their final syllable and stress pattern is likely to provide a satisfactory account. On the other hand, it is interesting to know if inclusion of possibly irrelevant information can be detrimental. In this study, we will test models in which up to four syllables are coded, both with and without stress information.

3.4.2 Comparability: Features and Alignment

In MBL models, inter-item comparability is based on features. Each exemplar has a value assigned to each feature and the distance between two exemplars is defined as the sum of the distances between corresponding feature values.

Clearly, any useful comparison of the phonology of exemplars has to involve features that are coded below the level of syllables. Figure 1 illustrates the feature representations that are compared in this study. The *onset-nucleus-coda* representation divides a syllable in three elements: the phoneme with maximal sonority (the nucleus), the phoneme(s) preceding it (the onset), and the phoneme(s) following it (the coda). This alignment method is commonly used in memory-based learning and is considered to produce a well-balanced representation. While all syllables have a nucleus, it is possible to have syllables without onset or coda. However, these "empty" feature values do count in the computation of similarity, so that two syllables with no value for the onset feature are considered fully similar with regard to that feature. It is not clear if empty feature values actually distort similarity, and this study does not try to address this issue. On a more practical level, we will compare the onset-nucleus-coda alignment method with a method that deals with empty feature values in a consistent manner.

For instance, the word /aːp/ (‘ape’), in which the first phoneme is also the one with maximal sonority, is represented as /=/, /aː/, /p/ with onset-nucleus-coda alignment (the ‘=’ symbol indicates that there is no value for a particular feature, in this case the onset). In *start-peak-end* alignment, the value of the nucleus feature is also used as a virtual value for onset and coda if no ‘real’ value is available. Hence, with start-peak-end alignment, the word /aːp/ is represented as /aː/, /aː/, /p/. A third alignment method that will be added to the comparison is an extension of start-peak-end alignment: *peak and valley* alignment uses the element with minimal sonority to divide a syllable’s onset and coda analogically to how start-peak-end divides the syllable by maximal sonority. For instance, onset-nucleus-coda alignment of the monosyllabic form /strant/ (‘beach’) would yield the features /str/, /a/, and /nt/. In peak and valley alignment the onset /str/ is further decomposed into its start, sonority valley, and end, giving the features /s/, /t/, and /r/. Likewise the coda is split further into its start /n/, its sonority valley /t/, and its end /t/. The final peak and valley representation of the syllable /strant/ consists of the 7 feature values /s/, /t/, /r/, /a/, /n/, /t/, and /t/.

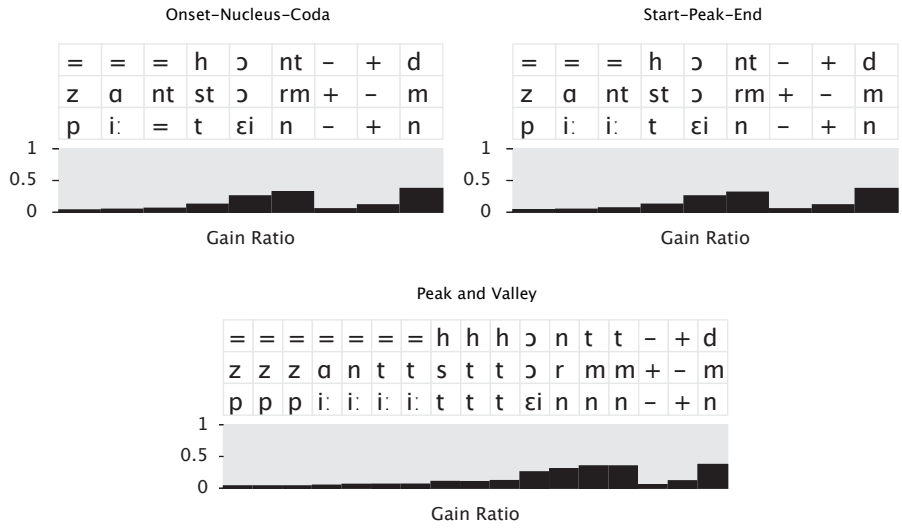

FIGURE 1: Examples of feature representations for the words /hɔnt/ (dog), /ˈzɑnt-stɔrm/ (‘sandstorm’), and /ka-piː-ˈtɛin/ (‘captain’). All representations use only the two final syllables of the words. The ‘=’ symbol indicates that there is no value for a particular feature. The last three values in each example indicate the presence of stress on the penultimate and final syllable, and the final grapheme of the word. Gain ratios obtained in the simulation studies are shown for each representation.

On top of the above methods that align phonological information within syllables, syllables must be aligned within words. Given a memory with exemplars with varying numbers of syllables, two decisions must be made. First, a choice must be made for word-final or word-initial alignment. Since the relevant information for Dutch noun plural inflection is mainly concentrated at the end of the word, a word-final alignment will be used. Secondly, exemplars consisting of fewer syllables than those required by the representation template of the model must be padded up with values for the features of missing syllables (leftward, in the case of word-final alignment, or rightward, for word-initial alignment). Two padding methods will be compared here. *Empty padding* uses one arbitrary value for all missing features. For comparison, we will use the *delta padding* method, which uses virtual values to refer to the values of the preceding syllable (right to left). For instance, a disyllabic onset-nucleus-coda representation would consist of 3 feature values for each of the syllables. With empty padding, the monosyllabic word /strant/ would have the feature values /=/, /=/, /=/, /str/, /a/, and /nt/. With delta padding, the empty slots are filled up with pointers to the next syllable and the resulting feature values are />str/, />a/, />nt/, /str/, /a/, and /nt/.

3.4.3 Similarity: Feature Weights and Distance Metrics

Feature Weights: In building an MBL model, we can, to a certain extent, exclude what we think is irrelevant information. However, there may be degrees of relevance for the information included in the model. For example, in Dutch noun plural inflection, it is probable that the features of the final syllable are more informative than the features of the preceding syllables. In memory-based learning, it is common to weight features by their information-gain with respect to the classification. $H(C)$ (Equation 1) is the entropy of the set C of class labels.

$$(1) \quad H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$$

The weight for a particular feature can then be defined as in Equation 2, where V_i is the set of values for feature i, and $H(C|v)$ the entropy of a value distribution over the different classes.

$$(2) \quad w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C|v)$$

In this study, we will use the gain ratio method, which normalizes information gain for the number of values per feature. In Figure 1, gain ratio feature weights are shown for some of the feature alignment methods compared in this study.

Distance metrics: We have defined the distance between two exemplars as the weighted sum of their by-feature distances, but we have not yet defined how feature value matching is computed.

In memory-based learning, the most straightforward method of assessing similarity is by the *overlap* distance: identical feature values have an overlap distance of 0, non-identical feature values have a distance of 1. Equation 3 gives the weighted overlap distance between two exemplars.

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

For numeric feature values, the absolute value of the normalized difference between the values is taken.

A consequence of using the overlap distance metric is that exemplars that do not overlap on any feature are at the same, maximal, distance (which is equal to the number of features in the case of unweighted features). Another consequence is that an exemplar may have many neighbors at the same distance. As we will see later, this has important consequences for setting the parameters of the decision function. A third characteristic of the overlap metric is that it does not allow for gradient similarity between feature values. For instance, given an onset-nucleus-coda coding of phonological information, the word *beak* (/b/, /iː/, /k/) has the same overlap distance (1) from both *peak* (/p/, /iː/, /k/) and *weak* (/w/, /iː/, /k/), although *beak* and *peak* are phonologically more similar than *beak* and *weak* are. Therefore, MBL models are often implemented using the Modified Value Difference Metric (MVDM) (Cost and Salzberg 1993), which provides gradient similarity for feature values. MVDM looks at co-occurrences between feature values and target classes. Feature values are considered similar if they have similar distributions over target classes. This is shown in Equation 4, where the inter-value distance (to be used in Equation 3) is a function of the conditional distribution of classes given the feature values.

$$(4) \quad \delta(v_1, v_2) = \sum_{i=1}^{n} |P(C_i|v_1) - P(C_i|v_2)|$$

Because the MVDM metric implements gradient similarity, the number of neighbors that are at the same distance from any given exemplar decreases dramatically relative to the overlap metric. This is an important factor when choosing the parameters of the decision function, which is the topic of the next section.

3.5 Decision

Once we have established which exemplars are in the model's knowledge base, how they are represented and how similarity between them is computed, a final

and equally crucial question concerns the nature of the decision function, i.e., how a class is assigned to novel exemplars given its similarity to each exemplar in memory.

3.5.1 Neighbors and Distance

A problem with the nearest neighbor approach is that several exemplars may be equally similar to a target exemplar. In that case, there may be several neighbors at a given distance. Rather than choosing k of these neighbors randomly, we use all neighbors at the same distance. Therefore, the parameter k should be interpreted as the number of nearest distances rather than as the number of nearest neighbors, and, even at k=1, several neighbors may be selected for extrapolation.

The most straightforward decision method is to base the class of a new exemplar on the class of the exemplar(s) at the nearest distance. Although quite successful for some problems, the 1-NN approach is mostly suitable for discrete classification tasks: if there is only one exemplar at the nearest distance, the method cannot provide a probabilistic output for different target classes. Furthermore, the 1-NN approach assumes that more distant exemplars are all equally irrelevant. For models dealing with linguistic productivity, such an assumption may be inappropriate because it fails to account for class size (type frequency) effects.

Another relevant consideration when setting a value for k is that the number of exemplars at a given distance is highly dependent on the distance metric. Compared to the overlap metric, the MVDM metric, which computes graded similarity between feature values, lowers the probability of finding equally distant exemplars.

3.5.2 Distance Weighting

Distance weighting reflects the intuition that the more distant a neighbor is from the target exemplar, the lower its influence is on the classification of that exemplar. In practice, distance weighting becomes more important with higher values of k, as more distant exemplars may jointly influence classification. We will compare *zero decay* distance weighting, in which each exemplar is equally weighted, with *inverse distance decay* weighting, where support of each neighbor is inversely proportional to its distance from the target exemplar.

3.5.3 Type Merging

When the memory of an MBL model contains identical forms with the same inflectional pattern, these forms are normally counted as distinct exemplars by the decision function. Because exemplar representations do not always correspond to the full word (e.g., due to the limit on the number of coded syllables),

the probability of having two identical forms is higher than in normal language. In some cases, especially with low values for k, this leads to a neighborhood largely composed of identical exemplars. For this reason, we compared the effect of counting all identical forms separately to that of "merging" them and counting only once.

3.5.4 Output

Instead of a classification, an MBL model can also give probabilities for different classes. As classification (except in the case of 1 neighbor) involves a probability distribution for each class, a model can be read-out at the pre-decision level. This probability distribution is local, however. MBL is a non-parametric approach that does not make assumptions about the global distribution of classes.

4. RESULTS AND DISCUSSION

For each of the three tasks, we ran 23,040 different simulations. Each simulation had a unique combination of values for the parameters listed in Table 1. Simulations with the overlap metric were run with k = 1,3,5, and 7. Simulations with the MVDM were run with k = 1,3,5,7... up to 51.

For the two pseudo-word tasks, a prediction was considered accurate if the simulation assigned a probability ≥ 0.5 to the answer given by the majority of human subjects. In the lexical reconstruction task, a prediction was considered correct if the simulation assigned a probability ≥ 0.5 to the lexically attested form.

In general, surprisingly good accuracy scores were observed. For comparison, the baseline accuracy (choosing the majority form, *-en*) was about 63 % in the lexical reconstruction task, and 68.75% and 62% in the first and second pseudo-word tasks respectively. In the lexical reconstruction task, the best simulation had an accuracy of 97.8%. For the first pseudo-word task, the best simulation was 100% accurate. The best simulation for the second pseudo-word task scored a fairly high 89% accuracy. For all tasks, a surprising number of outliers were observed towards the lower end of the scale, with some simulations achieving no more than 50% accuracy. Figure 2 shows that these outliers correspond to simulations where only one syllable was used in the exemplar representation.

Disregarding one-syllable simulations, minimal accuracy was 83.9% for the lexical reconstruction task, and 77.5% and 73.9% for the first and second pseudo-word tasks respectively. Table 1 gives accuracy scores on each task, with the exclusion of one-syllable simulations. Due to the large number of

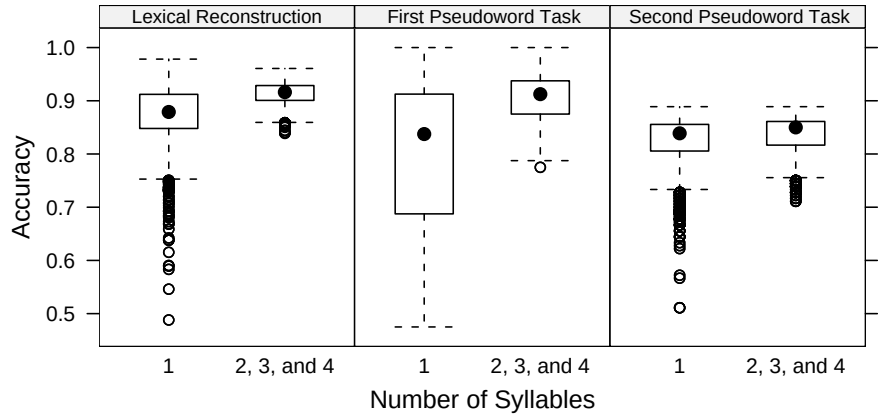

FIGURE 2: Box and whisker plots comparing the accuracy distribution of one-syllable simulations with two-, three-, and four-syllable simulations. Filled black dots indicate the median; box height shows the interquartile range Q3-Q1; the whiskers extend to the most extreme data point within 1.5 times the interquartile range. Points beyond the whiskers can be considered outliers in a normal distribution and are plotted separately.

data points analyzed, even very small differences between tested parameter values proved to be significant.[4]

4.1 Information and Representation

4.1.1 Number of Syllables, Stress, and Final Grapheme

As stated above, accuracy is clearly affected by the number of syllables used to represent exemplars. Even when one-syllable simulations are disregarded, some differences remain. In the lexical reconstruction task and in the second pseudo-word task there is a slight decrease in accuracy with increasing number of syllables. In the first pseudo-word task, on the other hand, an increase in the number of coded syllables is accompanied by a marked increase in accuracy. A possible explanation for this is that the first pseudo-word task included some stimuli that specifically benefit from analogies with three and four syllable words. Whereas no stimuli in the second pseudo-word task have more than two syllables, about 1 in 3 stimuli in the first pseudo-word task have three or more syllables. However, this does not fully explain why, in the lexical reconstruction task, the same proportion of items with more than two syllables are best predicted with a two-syllable representation of exemplars.

[4] The data used in this study (lexicon, pseudo-word stimuli), complete results, and analysis are available at `http://www.cpl.ua.ac.be/data`.

MEAN ACCURACY (STANDARD DEVIATION)			
	Lexical Reconstruction	Pseudoword Task1 (Baayen *et al.* 2002)	Pseudoword Task2 (Keuleers *et al.* 2007)
Number of Syllables			
2	.914 (.020)	.878 (.033)	.841 (.031)
3	.914 (.019)	.917 (.043) ***	.840 (.032)
4	.912 (.019) ***	.922 (.042) ***	.838 (.033) ***
Stress			
No	.915 (.019)	.898 (.041)	.837 (.032)
Yes	.912 (.020) ***	.913 (.046) ***	.842 (.032) ***
Final Grapheme			
No	.907 (.021)	.884 (.036)	.838 (.037)
Yes	.920 (.015) ***	.927 (.041) ***	.842 (.026) ***
Features			
Onset-Nucleus-Coda	.913 (.020)	.909 (.046)	.843 (.031)
Start-Peak-End	.913 (.019)	.903 (.044) ***	.836 (.037) ***
Peak and Valley	.915 (.018) ***	.905 (.042) ***	.840 (.027) ***
Padding			
Empty	.914 (.019)	.908 (.045)	.841 (.032)
Delta	.913 (.019) ***	.904 (.043) ***	.838 (.032) ***
Distance Metric			
Overlap	.927 (.018)	.873 (.037)	.844 (.028)
MVDM	.911 (.019) ***	.911 (.043) ***	.839 (.033) ***
Distance Weighting			
Zero Decay	.908 (.020)	.907 (.042)	.839 (.033)
Inv. Distance Decay	.919 (.016) ***	.905 (.046) ***	.840 (.031) ***
Class Labels			
Categorical	.927 (.010)	.900 (.044)	.821 (.030)
Transformation	.900 (.017) ***	.910 (.043) ***	.858 (.021) ***
Type Merging			
No	.913 (.019)	.906 (.044)	.839 (.032)
Yes	.914 (.019) ***	.906 (.044)	.840 (.032) *

TABLE 1: Mean accuracy and standard deviation for 17,280 simulations on three tasks. Values correspond to the average accuracy of all simulations with the parameter value specified in the left column. Asterisks indicate a significant difference with the first specified value of the parameter (*** = p<.001, ** = p<.01, * = p<.05)

There is a positive effect of including word stress in the two pseudo-word tasks, whereas the effect is slightly negative for lexical reconstruction. Inclusion of the final grapheme in the representation yields a significant increase in performance and robustness on all tasks. Since Dutch spelling is morphological, the final grapheme can hold information about the realization of the inflected form. For example, the form /hɔnt/ ('dog') is spelled *hond*, with its final grapheme indicating that the final phoneme is voiced in the plural /hɔndə/ (spelled *honden*). Another advantage is that the final grapheme may result in disambiguation of some phonological transcriptions in CELEX – which is based on a written corpus. For instance, as most Dutch speakers do not pronounce the final *n* in words such as *wagen* ('car'), the phonological rendering /waːɣə/ rhymes with words such as *sage* ('saga'), pronounced as /saːɣə/. While words of the *wagen* type almost invariantly take the *-s* suffix in the plural, words of the *sage* class do not show a clear preference for either plural suffix. Although phonological transcription in CELEX does not encode a pronunciation difference in the two word classes, there may still be a significant difference in their phonetic realization (Ernestus and Baayen 2004), which could justify including the final grapheme as a relevant disambiguating cue.

4.1.2 Feature Representation and Padding

In the lexical reconstruction task, simulations with *peak and valley* representation present a slightly higher accuracy than simulations with the baseline *onset-nucleus-coda* representation. In both pseudo-word tasks, the onset-nucleus-coda representation has a higher accuracy than the other two representations. For all tasks, the *empty* padding strategy obtains a slightly higher score than the *delta* padding method. In practice, the average differences between simulations on differently aligned and padded-up representations were so small that we can conclude that the more sophisticated methods do not give an additional advantage in these tasks. All in all, MBL appears to be fairly robust in the face of small differences in exemplar representation.

4.2 Distance Metric, k, and Distance Weighting

4.2.1 Results with the MVDM Metric

Figure 3 illustrates the interaction of the k parameter and the distance weighting function in the three tasks. The figure shows that the relation between k and accuracy is clearly non-linear. For the lexical reconstruction task maximal accuracy is reached with $k = 3$ and decreases thereafter. Decrease is less steep with the *inverse distance decay* weighting method than with the *zero decay* method. For both pseudo-word tasks, there is a steady increase in accuracy as k rises to a $\pm$ 5 value, followed by a plateau and a slow decrease for higher

values of k. Accuracy is maintained when the inverse distance decay method is used.

4.2.2 Results with the Overlap Metric

Simulations using the overlap metric yield a similar interaction with k. Maximal accuracy in lexical reconstruction is reached immediately at k=1 and decreases thereafter. In the first pseudo-word task, maximal accuracy is reached a bit later, at $k = 3$. In the second pseudo-word task, accuracy is still rising at $k = 7$, our final tested value. For all three tasks, the inverse distance decay method yields higher accuracies than the zero decay method.

Although we know from experience that the MVDM metric is particularly suitable for linguistic tasks, use of the overlap metric does not seem to badly affect accuracy. The magnitude of the lexicon may have played an important role here (see below).

Varying the k parameter has different repercussions on accuracy in lexical reconstruction vs the two pseudo-word tasks. For lexical reconstruction, the optimal value for k is near to one, while for the pseudo-word tasks k=1 is clearly aberrant, while values from 5 to 15 give better results. A possible explanation of this difference is that what needs to be modeled in a pseudo-word task is the true generalization capacity of the model. In the lexical reconstruction task, on the other hand, the goal is to make correct predictions for exemplars that may be produced differently through pure generalization. While we should be very careful about drawing general conclusions on the basis of this experimental evidence only, a possible consequence of this result is that lexical reconstruction does not provide a firm ground for stating generalizations about the nature of psycholinguistic processes.

The optimal values for k in the pure generalization tasks may also tell us something about inflectional morphology in general: The fact that a simulation with one or three nearest neighbors badly fits experimental data, may also be an indication that there is a lower bound on the class size of a productive inflectional process.

Use of the inverse distance decay weighting method alleviates the problem of diminishing accuracy for high levels of k observed with zero decay weighting. On the one hand, the inverse distance decay method is consistent with the view that even distant exemplars can influence the decision process. On the other hand, zero decay weighting makes the problem more tractable and allows us to see more easily when additional exemplars begin to have a damaging effect.

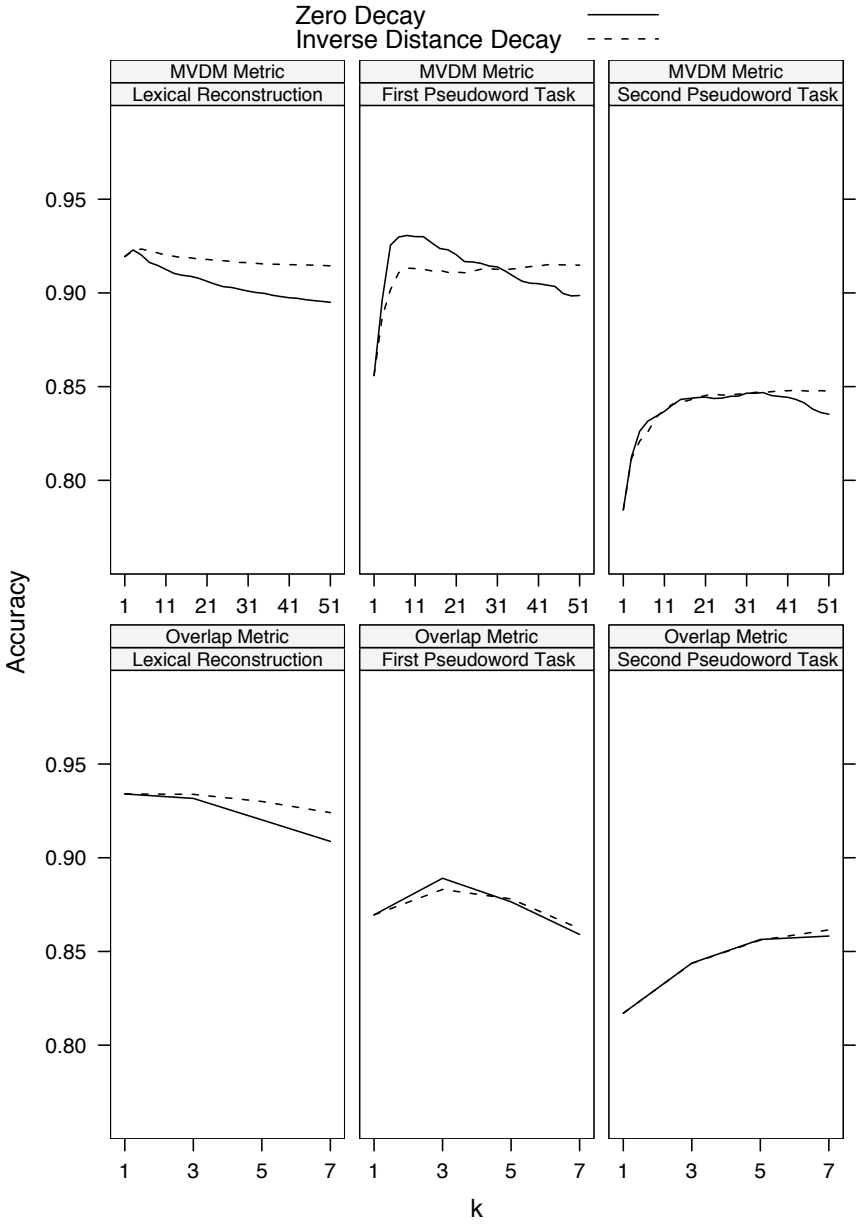

FIGURE 3: Mean accuracies for simulations by k, distance weighting method, and distance metric.

4.3 Class Labels and Type Merging

In the lexical reconstruction task, simulations with categorical labels perform better than simulations with transformation labels. In the pseudo-word tasks, on the other hand, simulations with transformation labels perform better than simulations with categorical labels.

There is a very small positive effect of type merging in the lexical reconstruction task and in the second pseudo-word task. In the first pseudo-work task, type merging does not have any effect. A possible explanation is that this is due again to the stimulus types used by Baayen *et al.* (2002). As type merging can only occur when a target exemplar has homophonic neighbors with the same inflectional pattern, this phenomenon is less likely to occur with a set of relatively complex stimuli.

Nature of the task is central to any classification problem. The use of categorical class labels (such as *en/s/other*) is only partially related to the inflected forms generated by human subjects. For instance, *-en* can occur with or without voicing of the final consonant, while the *other* label does not correspond to any specific transformation. Although high accuracy scores in class labeling are far from trivial, the results obtained by using more than 60 transformation labels are certainly more impressive. They show that memory-based learning models are able to deal with more complex issues in morpho-phonology. An interesting observation is that a transformation label only contains relevant information about the target form of the pair it was derived from. For instance, the transformation label derived from the singular-plural pair /hɔnt/-/hɔndə/ will specify one operation: 'substitute the final element of the source form by /də/'. This tells us that the original plural ended in /də/ but says nothing about the original singular. In theory, the transformation may apply to any form regardless of the phoneme it ends with. However, the only forms for which the transformation makes sense are source forms that end in a sonorant consonant + /t/. Applied to other source forms, the result is nonsensical in the context of Dutch plural inflection (e.g., /hɔnk/-/hɔnkə/, /kast-kasdə/). When we inspected the results of simulations, even average scoring ones, we found that errors in classification were the result of one sensible transformation being selected over another sensible transformation (e.g., *-s* instead of *-en*), but not of inappropriate transformations. With a set of over 60 transformation labels and a lexicon containing nearly 20,000 exemplars, this result is remarkable. It means that similarity appropriately constrains the exemplars selected for analogy and that no further restrictions are needed. For any target, close neighbors will always have transformation labels resulting in a sensible inflected form of the target.

Another noteworthy point is that categorical labels give better accuracy in the lexical reconstruction task while transformation labels fare better in the pseudo-word tasks. Because the definition of class labels interacts with other parts of the model, such as feature weighting and the distances obtained in

the modified value metric, the source of this disjunction is hard to pinpoint. Nonetheless, the results shown here suggest that transformation classes offer an effective alternative to labels based on a priori linguistic knowledge.

5. CONCLUSIONS

The simulations reported in this paper allowed us to take a closer look at Dutch noun inflection from different perspectives.

While classification accuracy was surprisingly high overall, detailed analysis of simulation results highlighted important differences among the three tasks. First, the two pseudo-word tasks and the lexical reconstruction task appear to require considerably different configurations of model parameters to yield optimal performance. The evidence, although non-conclusive, seems to suggest that observations concerning the psycholinguistic processes involved in lexical reconstruction tasks should be considered with great care. Second, the mean accuracy in the first pseudo-word task was about 10% higher than in the second pseudo-word task. A possible explanation for this difference is that the experimental results for the second set of pseudo-words were obtained through an experiment that deliberately aimed to skew the distribution of plural suffixes through the manipulation of word spelling. Pseudo-words were presented auditorily but were simultaneously visually presented in a typically Dutch spelling or a typically English spelling. A third condition did not show any spelling at all. Participants used the *-s* suffix more often in the English spelling condition than in the two other conditions, most likely through the association of the English spelling with loanwords, which have a preference for the *-s* plural in Dutch. Although the results from the English spelling condition itself are not included here, there may have been some crossover effects between conditions. If we accept that the distribution of responses may have been slightly skewed, it is a good sign that no simulation on this task resulted in very high accuracy by chance. Third, the mean accuracy reported here for the lexical reconstruction task was about 5% higher than the accuracy in a leave-one-out lexical reconstruction task on Dutch noun plural inflection also reported by Keuleers *et al.* (2007). This is surprising, because the leave-one-out test protocol, which uses the whole lexicon minus one exemplar to predict the class of that exemplar (repeated as many times as there are exemplars), is expected to give better results because a larger proportion of exemplar evidence is tapped for the task. However, it should be noted that Keuleers *et al.* (2007) used a smaller lexicon of monomorphemic nouns (3,135 exemplars) while in the simulations reported here the lexicon contained more than 19,000 word forms of arbitrary morphemic complexity.

The simulations also provided us with important insights into the robustness of MBL. With the only exception of one-syllable models, changing parameter values did not cause dramatic fluctuations in accuracy. A factor that

may have contributed to this robustness is lexicon size. In the machine language learning literature, the impact of different parameter values and even different machine learning methods is shown to decrease with an increasing size of training data (Banko and Brill 2001). The lexicon we used was very large (more than 19,000 items) compared to lexica used for other tasks in similar domains. If we adopted the same sampling criteria to create a lexicon for English past tense inflection, for instance, we would get a collection of about 2,000 items. Were a smaller lexicon used in our task, some of the minor differences we observed in this study could have been substantially larger.

In the computational modeling of psycholinguistic processes, it is important to know what the results of a simulation tell us about the process we are trying to model. This is crucially connected with how the parameters of that simulation were chosen and how well the simulation generalizes to other data. The standard practice in statistical or machine learning approaches to language processing is to carry out a lexical reconstruction task by systematically trying out different parameter settings. The best settings are then used on the target task and only results of that simulation are reported. As we argued above, the best performing simulations on the lexical reconstruction task turn out to have suboptimal accuracy for the pseudo-word tasks. It looks like optimal accuracy in lexical reconstruction is due to factors that are somewhat orthogonal to human generalization behavior.

It is not uncommon, in computational psycholinguistics, to run simulations with a wide range of parameter settings and report the results of the best performing simulation as the performance of the theoretical model under consideration. When a new task is addressed, a new set of simulations is run and, again, the best performing one, which may have been obtained with completely different parameter settings, is reported. In isolation, however, this optimal result may be quite misleading. Reporting the best outcome only tells us that the theory under consideration might be right, but not how hard it is to falsify it. There is no way of knowing what other outcomes have been predicted by simulations with different parameter settings, nor if the results of the best performing simulation are exceptional considering the results of the unreported simulations.

A first alternative is to consider only simulations within a limited range of parameter settings that are sensible based on expert knowledge of the task domain. Reducing the number of outcomes, this approach increases the significance of the best performing simulation. A drawback, of course, is that this precludes discovery of better performing simulations with parameter settings that were considered insensible beforehand.

A second alternative is to summarize the results of all simulations instead of reporting only the best performing one. If the performance range is known, then we also know how hard falsification is, and this gives an indication of the strength of the theory under consideration. Box and whiskers plots, such

as those in Figure 2, convey a great deal of information on the distribution of results (although it should be noted that the distribution of accuracies does not necessarily reflect the distribution of outcomes). In the case presented here, we see that, with the exception of one-syllable models, the distribution of classification accuracies indicates that a large number of simulations in fact cover a small portion of the solution space. With knowledge of this distribution the relevance of the best score can be more easily understood. Back to the box and whiskers plots, we see that, for all tasks, the best performing simulations would not be considered outliers in a normal distribution: Although many simulations with different parameter settings give worse results, the best performing simulations are unexceptional instances of MBL as a theory of inflectional morphology.

To conclude, we argue that a good methodology for computational psycholinguistics is to explore as many simulations as possible with different information sources (features), instance representations, class representations, and algorithm parameter settings, and to show transfer of good parameter settings for different psycholinguistic tasks. By using Dutch plural inflection as an example, we have shown that this approach is feasible and provides more insights both into the task and into the potential psychological relevance of MBL models.

REFERENCES

Albright, A. & Hayes, B. (2003). *Rules vs. analogy in English past tenses: A computational/experimental study*. «Cognition», 90(2), 119–161.

Baayen, R. H. (2001). *Word frequency distributions*. Dordrecht: Kluwer.

Baayen, R. H. & Moscoso del Prado Martín, F. (2005). *Semantic density and past-tense formation in three Germanic languages*. «Language», 81, 666–698.

Baayen, R. H., Piepenbrock, R., & Gulikers, L. (1995). *The CELEX Lexical Database [CD-ROM]*. Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania [Distributor].

Baayen, R. H., Schreuder, R., De Jong, N., & Krott, A. (2002). *Dutch Inflection: The Rules That Prove The Exception*. In S. Nooteboom, F. Weerman, & F. Wijnen (eds.), *Storage and computation in the language faculty.* Dordrecht: Kluwer.

Banko, M. & Brill, E. (2001). *Scaling to Very Very Large Corpora for Natural Language Disambiguation*. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. (pp. 26–33).

Bloomfield, L. (1933). *Language.* New York: H. Holt and Company.

Bybee, J. L. (1995). *Regular Morphology and the Lexicon*. «Language and Cognitive Processes», 10, 425–455.

Clahsen, H. (1999). *Lexical entries and rules of language: a multidisciplinary study of German inflection*. «Behavioral and Brain Sciences», 22(6), 991-1013; discussion 1014–60.

Cost, S. & Salzberg, S. (1993). *A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features*. «Machine Learning», 10, 57–78.

Croft, W. & Cruse, D. A. (2003). *Cognitive linguistics*. Cambridge: Cambridge University Press.

Daelemans, W. (2002). *A Comparison of Analogical Modeling of Language to Memory-based Language Processing*. In D. Skousen, R. Lonsdale, & D. Parkinson (eds.), *Analogical Modeling*. (pp. 157–179). Amsterdam: John Benjamins.

Daelemans, W., Gillis, S., & Durieux, G. (1994). *The Acquisition of Stress: A Data-oriented Approach*. «Computational Linguistics», 20, 421–451.

Daelemans, W. & van den Bosch, A. (2005). *Memory-Based Language Processing.* Cambridge: Cambridge University Press.

Daelemans, W., Zavrel, J., van der Sloot, K., & van den Bosch, A. (2007). *TiMBL: Tilburg Memory Based Learner, version 6.0, Reference Guide. ILK Technical Report Series*, 07–05.

De Haas, W. & Trommelen, M. (1993). *Morfologisch handboek van het Nederlands. Een overzicht van de woordvorming. (Handbook of Dutch Morphology. An overview of Word Formation).* 's-Gravenhage: SDU.

De Saussure, F. (1916). *Cours de linguistique générale.* Lausanne – Paris: Payot.

Eddington, D. (2000). *Analogy and the dual-route model of morphology*. «Lingua», 110, 281–289.

Ernestus, M. & Baayen, R. H. (2004). *Analogical effects in regular past tense production in Dutch.* «Linguistics», 45(5), 873–903.

Estes, W. K. (1994). *Classification and cognition* . Oxford: Oxford University Press.

Fix, E. & Hodges, J. L. (1951). *Discriminatory analysis. Nonparametric discrimination: consistency properties. [Technical Report].* Randolph Field, Texas: USAF School of Aviation Medicine.

Gillis, S., Durieux, G., & Daelemans, W. (2000). *Lazy Learning: Natural and machine learning of word stress*. In P. Broeder & J. Murre (eds.), *Models of Language Acquisition*. (pp. 76–99). Oxford: Oxford University Press.

Hahn, U. & Nakisa, R. C. (2000). *German Inflection: Single Route or Dual Route?* «Cognitive Psychology», 41, 313–360.

Keuleers, E., Sandra, D., Daelemans, W., Gillis, S., Durieux, G., & Martens, E. (2007). *Dutch plural inflection: The exception that proves the analogy*. «Cognitive Psychology», 54(4), 283–318.

Krott, A., Schreuder, R., Baayen, R. H., & Dressler, W. U. (2007). *Analogical effects on linking elements in German compounds*. «Language and Cognitive Processes», 22(1), 25–57.

Krott, A., Schreuder, R., & Baayen, R. H. (2002). *Linking elements in Dutch noun-noun compounds: constituent families as analogical predictors for response latencies*. «Brain and Language», 81(1-3), 708–722.

Nosofsky, R. M. (1988). *Similarity, frequency and category representations*. «Journal of Experimental Psychology: Learning, Memory and Cognition», 14, 54–65.

Pinker, S. (1999). *Words and Rules.* London: Phoenix.

Ratcliff, J. W. & Metzener, D. E. (1988). *Pattern Matching: the Gestalt Approach*. «Dr. Dobbs Journal», pp. 46–51.

Rumelhart, D. E. & McClelland, J. L. (1986). *On Learning the Past Tenses of English Verbs*. In J. L. McClelland, D. E. Rumelhart, & The-PDP-Research-Group (eds.), *Parallel Distributed Processing. Explorations in the Microstructure of Cognition: Vol. 2. Psychological and Biological Models*. (pp. 216–271). Cambridge, MA: MIT Press.

Skousen, R. (2002). *An overview of analogical modeling*. In R. Skousen, D. Lonsdale, & D. Parkinson (eds.), *Analogical Modeling*. (pp. 11–26). Amsterdam: John Benjamins.

van den Bosch, A. & Daelemans, W. (1999). *Memory-based Morphological Analysis*. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, ACL '99, 285–292.

SUMMARY: Il presente lavoro indaga il paradigma del "memory-based learning" (MBL) inteso come un modello capace di riprodurre il comportamento linguistico della flessione plurale del nome in olandese. Dapprima delineeremo l'origine e i riferimenti teorici dell'approccio MBL per fornire, in seguito, una breve panoramica della flessione plurale dei nomi in olandese e una descrizione dettagliata dell'uso dei modelli MBL per la morfologia flessiva. I risultati di un'ampia serie di simulazioni su tre compiti di flessione del nome plurale verranno analizzati in dettaglio. In particolare, illustreremo gli effetti differenti legati al variare delle configurazioni di parametri dell'algoritmo del MBL, ai problemi di rappresentazione degli esempi e alle differenti definizioni della flessione intesa come compito di classificazione. Nella parte finale, tali risultati saranno considerati in relazione alle correnti procedure per l'ottimizzazione dei parametri del modello e per l'analisi e la valutazione dei risultati delle simulazioni.