Minimal generalization of Dutch diminutives

Tim Dorscheidt,[a,†] Nikola Valchev,[a,†]Terence van Zoelen[b,†], and John Nerbonne[b,†]

[a]*School of Behavioral and Cognitive Neurosciences, University of Groningen*

[b]*Department of Humanties Computing, University of Groningen*

Abstract

A comparative study was conducted about the acquisition of diminutive forms in the Dutch language. A former study, using the C4.5 algorithm, is discussed and contrasted with the implemented Minimal Generalization model by Albright and Hayes. In addition, the model is also compared to a conducted behavioral study using wug words that follow the Dutch phonetic rules, but do not exist in the language. The Minimal Generalization model is very good in creating the correct diminutive forms from lemmas. In addition, the model corresponds quite well with the behavioral data. A notable exception are the wug words intended to correspond to the rule with suffix *kje*, for them the Minimal Generalization model does not give good estimates. An explanation for this result is given. The authors believe that the model's method of learning the necessary rules for this task displays characteristic similarities to the way humans learn these rules.

† Correspondence should be send to T.Dorscheidt@student.rug.nl, N.Stanimirov@student.rug.nl, T.van.Zoelen@rug.nl or j.nerbonne@rug.nl

1. Introduction

One of the big debates on language acquisition concerns the question whether humans have innate knowledge for learning language (for a recent summary see Yang, 2004). Scientists in favor (see Chomsky, 1965) argue that it is not possible to learn certain aspects of language without such innateness, whereas others believe the received input has enough information to allow extraction of all necessary knowledge (for discussion on grammatical class extraction, see Mintz et al., 1995).

This debate has been going on for decades, and is far from being decided. A strong argument in favor of innateness is the argument from the poverty of stimulus (Chomsky, 1980; Crain et al., 2001). It is believed that the language exposure children receive is not enough to explain some aspects of adult language. Complicated sets of rules and their exceptions are deemed impossible to learn from the available input, either because it does not contain any instances from which to acquire the knowledge, or it does not contain enough of them. Studies on the actual input children receive are difficult and often inconclusive, but seem to indicate that the argument from the poverty of stimulus is not as strong as widely assumed (Pullum et al., 2002).

A more direct and powerful technique to see whether natural exposure to language contains enough information is the use of computational models. If a "naïve" learner is capable of extracting the necessary classifications or rules, then innateness is not critical. It is currently impossible to construct a full computational implementation of a system capable of learning a natural language, therefore small parts of language are tested with narrow models. The model used in this paper will represent such a naïve learner, which will not have any a priori knowledge on the rules that need to be learned. Some feature extraction knowledge will be built in, to allow the model to interpret the input-data and learn from it. It is therefore not a study aimed at disproving or weakening innateness as a theory of language acquisition, but the following paper will test an already available model that could prove useful for doing just that.

The learning problem at hand is the acquisition of rules necessary to form Dutch diminutives from noun-lemmas. A study specifically targeted at the problem of learning Dutch diminutive forms, and using machine learning to do so, has been performed by Daelemans et al. (1997). This research will be discussed briefly and offers some interesting comparisons. An explanation of the theory behind the chosen model will follow, and it will be argued why this model is preferable to the alternative by Daelemans *et al*. A parallel survey-study provides data about the behavior of native speakers when using Dutch diminutives, enabling a comparison between the implemented method and the natural system it is trying to model. But, for those unfamiliar with Dutch diminutives, a short introduction on this common structure in Dutch language will start off the paper.

1.1. Dutch diminutives

A Dutch diminutive is the inflected form of a noun (other grammatical categories are possible too, but they do not have diminutive forms by standard and will be excluded in this study), usually changing the meaning of the uninflected word to something smaller. For example, *tafel* ('table') becomes *tafeltje* ('small table') in its diminutive form (for more reading, see Trommelen, 1983).

The standard rule for making Dutch diminutives is adding '*tje'* to the base of a noun. In general there are five known suffixes for the Dutch diminutive form, these are *tje*, *je*, *pje*, *kje*, and *etje*.

The frequency distribution of these five suffixes in the CELEX database (Max Planck Institute for Psycholinguistics, Nijmegen) is shown in table 1. These results are drawn from 3889 unique diminutive nouns.

Table 1. Frequency per diminutive form.

[tjə]	1879	48.3
[jə]	1452	37.3
[pjə]	102	2.6
[kjə]	76	1.9
[ətjə]	370	9.5
Exceptions	10	0.2

To give a quick introduction to the known phonological rules for diminutive forming, we list the general rules found by the machine learning application used by Daelemans et al. (1997) in their study:

- [jə] is used after an obstruent like [pɔpjə] (‘small doll’).
- [pjə] is used after a long vowel, diphthong or schwa, followed by [m], like in [beːzəmpjə] (‘small broom’). [pjə] is also used after a short vowel followed by a liquid ([r] or [l]) plus [m].
- [ətjə] is used after a nasal ([m], [n] or [ŋ]) or the liquid [l] preceded by a short vowel (as in: [roːmɑn-ətjə] (‘short novel’), [bɑl-ətjə] (‘small ball’)). This diminutive ending is also added after monosyllabic words with a final [r] that is preceded by a short vowel like in ([bɑr]).
- [kjə] is used in multisyllabic words ending in [ɪŋ] (like [koːnɪŋ]) if the stress is on the penultimate syllable, like in [sɔlderɪŋ-kjə] (‘small pieces soldered together’). The rule is competes strongly with the rule for [ətjə], for example in words like [leːrlɪŋ] (‘student’) en [tweːlɪŋ] (‘twin’), which are both ending on [ətjə].
- [tjə] is the default rule if none of the above apply.

Some words can have more than one suffix. For a more detailed discussion on this topic, see Daelemans et al. (1997). This paper only discusses some of these rules. The general rule for words having two syllables is: if they contain a short vowel, and the first syllable is stressed, and the second syllable has a nasal or a liquid [l], then the following rules can be applied:

- If a word ends on [n] or [l]; [tjə] and [ətjə] are both possible.
- If a word ends on [m]; [pjə] and [ətjə] are both possible.
- If a word ends on [ŋ]; [kjə] and [ətjə] are both possible.

For monosyllabic words, the following rules apply for multiple diminutive suffixes:

- If a word ends on [p] [b] or [ɣ], [ətjə] and [tjə] are both possible.
- If a word ends in a long vowel followed by a sonorant after [m], [tjə] and [ətjə] are both possible.

1.2. Dutch diminutive learning by C4.5

A study by Daelemans et al. (1997) on Dutch diminutives applied the C4.5 algorithm. C4.5 is a descendant of the program ID3 (Quinlan, 1987). It can perform a classification task on attribute-valued objects (the data) by supervised learning, which means that the categories must have been established beforehand. The classes must have been designed in a way that every single case can be assigned to only one specific class.

The C4.5 program generates a decision tree with leaves and decision nodes. Every leaf corresponds to a class and a decision node specifies some test on a single attribute of an object. Every outcome of the test leads to a one branch in the sub tree.

As explained by Quinlan (1987), the learning algorithm starts by receiving a collection of attributed objects with an pre-assigned class as input. In Daelemans’ case the objects were diminutive forms attributed with phonological features, and each object was assigned to the correct diminutive suffix as its class. With this the algorithm is able to construct a decision tree with leaf nodes and decision nodes with the provided information.

The building process and positioning of the nodes in the tree is done by calculating the minimal description length (Quinlan, 1989), which is done by choosing the best rules on the basis of the attributes. The construction of the tree is performed by applying a recursive method, which is included in appendix A.

Once the tree is constructed, the readability of the tree can be improved by pruning the obtained rules (appendix A). The rules are then converted into a readable table where the rules are sorted in a logical order, after which the algorithm can be tested on new input forms.

1.3. Dutch diminutive learning by minimal generalization

Numerous applications of machine learning algorithms can be found in the area of natural or artificial language learning. The studies can include or lack initial bias and can be based on human learning methods or not. Human-inspired methods benefit from plausibility, but it is far from clear how humans learn in general or linguistically. The previously described C4.5 cannot be considered a correct approximation of the actual human learning of the Dutch diminutive forms. It needs to have all its classes predefined, and such specific knowledge is unlikely to be innate. In addition, the enormous overhead requirements make the algorithm an even more implausible explanation of the technique used by humans. Storing all possible nodes and trying out numerous new nodes on the basis of high entropy for each and every new input is also unlikely at best, not to mention the regular pruning necessary to keep the decision-tree efficient.

There has been a recent addition to the field of learning algorithms, based on presumed human intuitive and/or

stochastic learning. Minimal generalization is an implementation based on three human inspired criteria, with the goal of mimicking human learning of phonological and morphological rules in a natural language (Albright & Hayes, 2002). The first criterion is the generation of complete output forms. The model should be capable of giving a genuine answer as a human would be expected to give, and not an abstract classification. Secondly, if possible, the model should give more than one answer with an estimate of its appropriateness, for humans can typically provide multiple answers as well and say how good they are. The third criterion is the possibility for a model to pick up variations in language patterns on a detailed level of distinction. When generalizing rules (this will be explained shortly) the model should note small irregularities and learn these as well, in the same way humans are capable of learning small variations on common language constructs as exceptions.

The developers of minimal generalization learning deem these criteria necessary in order to begin mimicking the way a human learns to generalize rules for a language. The advantages in comparison to an algorithm such as C4.5 are not only in the way it mimics humans better. The overhead is reduced because the minimal generalization learner (MGL) has a constant pruning capacity by way of statistically tracking the applicability of each rule (more on this later). Furthermore, by having competing generalized rules that generate multiple guesses the model is kept flexible and capable of continuous adjustment to the training input. A short explanation of how the model works will make matters more clear, but for a detailed inquiry into its workings, please read Albright & Hayes 2002 or 2003.

The algorithm starts by receiving pairs of pre- and post-processed forms of input, usually a lemma and its derived form (for instance English past tense inflection (Albright & Hayes, 2002 and 2003)), both of them encoded in the corresponding phonetic form. For each of the forms the model needs to know the salient phonetic features, which is the minimal available information on which (presumably) humans too can base their learning. These features enable the model to note all the changes that take place within pairs, on the basis of which the model can then start to form generalizations. Applying MGL to diminutive forms, here is an example for the Dutch nouns *roos* (‘rose’) and *kaas* (‘cheese’).

(1)
$$\emptyset \longrightarrow j\text{ə} \;/\; \#r \qquad o\text{:} \qquad s\_\#$$
$$\emptyset \longrightarrow j\text{ə} \;/\; \#k \qquad a\text{:} \qquad s\_\#$$

These rules (1) show that the model simply learned to add *je* ([jə]) after the lemma in order to get the correct diminutive form, which was the minimal change observable (from nothing, to *je*).

These initial learning steps enable the model to learn for each pair the minimal change between the lemma and its derivative. However, it would be pointless to only remember this for all input pairs. The model is therefore equipped with further generalization capacities. When receiving the input for *kaas* after *roos* it will not immediately create the rule mentioned in (1), but first try to apply already existing rules. With only the rule expecting *roos* as a lemma it will fail, but it can generalize if there are common features, leading to a new rule (2).

(2)
$$\emptyset \longrightarrow j\text{ə} \;/\; X \begin{bmatrix} +syllabic \\ +voiced \\ -tense \\ \dots \end{bmatrix} s\_\#$$

This rule is a generalization of the two earlier mentioned word-based rules. This rule will add [jə] after a word when it starts with a generic X (any set of phonemes), followed by a phoneme that has the minimally shared features of *o* and *a* (not all shared features are listed) and finally the *s* which is shared by both lemmas.

With many pairs of input, the model is capable of searching for such phonetic regularities in the changes made between pre- and post-processed training forms. It will create new rules that generalize with a minimum of common features. It is therefore called minimum generalization, and it is this kind of rule-forming that forms the basis of the model. With each new input pair, it searches whether it fits an already made rule, and if not, it copies the most appropriate rule to create a new one that does encompass this new occurrence. But, it does not remove the old rule, and this is an important part of the learning algorithm. The new rule is not necessarily more generally applicable, since the new input-form could be an exception. All rules need to defend their applicability by keeping track of their hits (the number of times the rule could create a correct output form) and their scope (the number of times the rule was applicable). These hits and scope are used to calculate reliability (hits/scope) and confidence (an adjusted reliability, taking absolute score into account, so that for instance 98% correct guesses for a scope of 500 gets a higher confidence then 100% correct for a scope of 2, see Mikheev, 1997). With this reliability or confidence, all applicable rules compete to generate output.

The model is now capable of constantly generating further generalizations on the basis of phonetic features in the input. It penalizes rules which generate incorrect output forms, increasing their scope without a hit (lowering reliability), as well as rules which do not generate output at all, failing to increase their scope (keeping confidence low).

One of the elegant behaviors expressed in this model is its indiscriminate attitude towards regular and irregular forms. The model will generate generalizations wherever it can and more regular forms will naturally lead to rules with a larger scope, but irregular forms will have their own specific similarities that enable the creation of a rule with a

high hit to scope ratio (the most exceptional cases having 1 hit to a scope of 1). Albright and Hayes claim that humans learn in a similar fashion, not making a fundamental distinction between regular and irregular forms, but only a generalization with a different scope.

The strong claim is that the MGL is capable of learning as humans do, intuitively and stochastically. This, in contrast to C4.5, is why it has become this study's model of choice to apply to Dutch diminutive learning. The model will receive Dutch word pairs of matching lemma and diminutive form, and hopefully learn to form the correct generalizations, whether of regular or irregular form.

2. Hypothesis

The C4.5 learner used by Daelemans et al. (1997) is making use of *a priori* knowledge about the domain of the language learning problem to apply the proper rule to the encountered examples. In the case of Daelemans's study the five grammatical Dutch diminutive categories were coded beforehand. In contrast the minimal generalization learner used by Albright and Hayes (2002) can perform a learning task without any domain specific categorical knowledge. With the minimal generalization learner it is possible to test the hypothesis that grammatical forms can be learned without prior knowledge of the formal rules of language.

To test the hypothesis, the outcomes of the minimal generalization learner can be compared with the results from the C4.5 learner and the behavioral data. If a correlation with the behavioral data can be found, the learner would seem to behave humanlike in applying language rules. We expect that the MGL will provide an equal rate of correct answers as the C4.5, indicating that the previous introduction of the grammatical classes is not needed for the production of the right answer.

3. Behavioral study

In order to be able to validate the predictions of the model, the same testing conditions were used for the MGL and proficient Dutch speakers. An unbiased way of testing people's way of forming the diminutive form of a substantive is using 'wug words'. Wug words are invented words that do not exist in a specific natural language, but do follow the phonetic rules of that language. For this test a set of 25 wug words was created in a way that for each of the five rules identified by Daelemans et al. (1997) there were applicable to five wug words. In this way it was possible to select the preferred form for the native Dutch speakers and to compare it with the most confident (the one that has highest confidence score) according to the Minimal Generalization Learner.

3.1. Method

Wug words were created using a program that uses the CELEX corpora and the phonotactic rules of the Dutch language in order to create strings of readable letters that are not actual words (for details on the program, see Duyck et al., 2004). A total of 128 wug words were created with 3, 5 and 6 letters. From them 25 were selected for the study (see table 2 for a list) on the basis of similarity to real Dutch words. The questionnaire consisted of 125 questions corresponding to all five possible forms of each wug word. Participants were required to give their confidence rating ("How well does this form sound to you?") on a scale from 1 to 7. The questionnaire was completed by students at the University of Groningen on a voluntary basis (for an example of the questionnaire see Appendix B)

3.2. Participants

35 students from the University of Groningen participated in the study: 20 females and 15 males. Participants' age ranged from 18 to 29 years (Mean = 20.4, SD = 2.24).

3.3. Results and Discussion

The ratings were analyzed to determine the discriminability between the diminutive forms. For each wug word a five level ANOVA was computed comparing the mean ratings for each of the five forms (See table 2).

The results showed that people do discriminate between the various diminutive forms for all but 7 of the wug words, for which there were two equally preferred diminutive forms.

4. The Minimal Generalization Learner

In this section the implementation of MGL will be briefly explained, and tested on CELEX-data and the wug words used in the behavioral test. The results will be discussed and a short comparison with the results of C4.5 will follow.

4.1. Method

The minimal generalization learner is this paper's model of choice for the diminutive learning problem. Its application is made possible by using Albright and Hayes' own Java code, generously supplied by the authors themselves. This MGL-program is capable of using a file as its input-source, and another file for determining which feature each phoneme has. The contents of the feature-translation file and a small part of the input-file are seen in Appendix C (appendix will be included in the full paper).

Table 2. A list of all 25 wug-words and the behavioral survey results. For each wug-word an one factor ANOVA was calculated with suffix as a 5 level independent factor. The table shows the preferred suffix for the behaviour data and the highest confidence and reliability suffix from the MGL.

Wug word	Phonetic form	ANOVA	Preferred diminutive forms	MGL highest confidence	MGL highest reliability
Bjarn	bjɑrn	F(4,170)=25.28; p<0.01	tje	tje (0.91)	tje (1)
Kwolm	kʊɔlm	F(4,170)=40.38; p<0.01	pje	pje (0.96)	pje (1)
Teppam	tɛpɑm	F(4,170)=46; p<0.01	etje pje		
Kerneung	kɛrnø:ŋ	F(4,170)=10.73; p<0.01	kje etje	tje (0.65)	tje (0.78)
Kjornang	kjɔrnɑŋ	F(4,170)=40.36; p<0.01	etje kje	etje (0.9)	etje (1)
Tuiem	tœyɛm	F(4,170)=32.48; p<0.01	pje etje	etje (0.85)	etje (1)
Keloem	kəlum	F(4,170)=51.05; p<0.01	etje pje	tje (0.65)	tje (0.78)
Augeng	ʌɯXɛŋ	F(4,170)=30.87; p<0.01	etje kje	tje (0.65)	tje (0.78)
Sterm	stɛrm	F(4,170)=35.04; p<0.01	pje	pje (0.96)	pje (1)
Jollang	jɔlɑŋ	F(4,170)=40.65; p<0.01	etje	etje (0.9)	etje (1)
Beklo	bəklo:	F(4,170)=28.43; p<0.01	tje	tje (0.96)	tje (0.98)
Nugerg	nʏɣɛrɣ	F(4,170)=18.04; p<0.01	je	je (0.92)	je (0.95)
Pliekung	plikuŋ	F(4,170)=27.23; p<0.01	etje	tje (0.65)	tje (0.78)
Yramp	ɛirɑmp	F(4,170)=24.82; p<0.01	pje	pje (0.91)	pje (1)
Tjerpvlirm	tjɛrpvlIrm	F(4,170)=18.57; p<0.01	pje	pje (0.96)	pje (1)
Pamuguls	pa:mʏɣɯls	F(4,170)=15.74; p<0.01	je	je (0.98)	je (1)
Rekstolm	rɛkstɔlm	F(4,170)=31.67; p<0.01	pje	pje (0.96)	pje (1)
Vaaft	va:ft	F(4,170)=47.31; p<0.01	je	tje (0.99)	tje (1)
Kanneil	kɑnɛil	F(4,170)=24.83; p<0.01	tje	tje (0.93)	tje (1)
Pailigoer	pailiɣur	F(4,170)=24.14; p<0.01	etje	etje (0.93)	etje (0.97)
Krihl	krIl	F(4,170)=17.92; p<0.01	je	etje (0.94)	etje (1)
Padelm	pa:dɛlm	F(4,170)=35.87; p<0.01	pje	pje (0.96)	pje (1)
Gemonk	Xəmɔnk	F(4,170)=21.74; p<0.01	kje	kje (1)	kje (1)
Konang	ko:nɑŋ	F(4,170)=23.13; p<0.01	etje kje	etje (0.9)	etje (1)
Quinsel	kʊInsɛl	F(4,170)=22.09; p<0.01	tje etje	tje (0.99)	tje (1)

4.2. Testing the MGL with CELEX-data

4.2.1. Introduction to CELEX

CELEX was founded in Nijmegen in 1986 under supervision of several Dutch-based research centers, most notably the Max Planck Institute for Psycholinguistics in Nijmegen. The project came to an end in 2001. The data is still available on CDROM and through a web interface. The database contains orthographic, phonological, morphological, syntactic and frequency properties of Dutch, English and German lemmas. For the Dutch language the database contained (in 1990), 381.292 Dutch word-forms, corresponding to 124.136 lemmas.

4.2.2. Obtaining the diminutive forms from CELEX

For this experiment we used the web based CELEX database. We abstracted 3869 Dutch Diminutive forms from it.

Table 3. Example input training data for the MGL[1]

Stem	Stem + Dim.	Freq.	Stem	Stem + Dim.
mɑɣə'zɛin	mɑɣə'zɛintjə	1	magazijn	magazijntje
'ze:rɛis	'ze:rɛisjə	4	zeereis	zeereisje
ɑtəl'je:	ɑtəl'je:tjə	2	atelier	ateliertje
we:və'rɛi	we:və'rɛitjə	1	weverij	weverijtje

The data in table 3 is used as training input for the minimal generalization learner. The first column is the stem in the phonological notation and the second column is the Dutch diminutive form. The third column is the frequency of the word form in the corpus from which the CELEX database is abstracted. The last two columns are the words in normal Dutch writing and primarily used as annotation of the data.

As already explained in a previous chapter, the MGL learns by using input pairs, in this case CELEX lemma-diminutive pairs. The program then uses the feature-file to derive phonetic regularities and derive the generalization rules. In addition, the MGL is capable of receiving bare input, the lemma form only, and giving all possible outputs with their confidences. To test the intrinsic learning capacity of the model, before comparing it with the behavioral data, we performed a ten-fold cross-validation. We randomly divided the entire CELEX-corpus into ten parts. Each part was then tested with the remaining 9 parts for learning, resetting the learned rules every time. This resulted in testing 3869 Dutch diminutive forms from the CELEX-corpus, deriving for each lemma all possible output-forms and their confidence and reliability ratings. To derive an overall score of the MGL's performance, the output forms with maximum confidence and reliability were compared with the correct form as contained within the CELEX-corpus.

To compare the MGL's performance with human behavior, the wug words presented to the human subjects were also tested. This time, the entire CELEX-corpus was used as training-data, after which the wug words were presented. All wug word-inputs generated one or more output forms with their respective reliability and confidence ratings. A comparison with the behavioral data can be seen in the results-section.

4.2.3. Results of MGL tested on CELEX

As mentioned the MGL was tested, in ten parts, with all available and appropriate input-forms from CELEX.

Table 4. Example output for [mɑɣə'zɛin].

Input	Output	Dim.	Scope	Hits	Rel.	Conf.
mɑɣə'zɛin	mɑɣə'zɛinjə	jə	3038	1250	0.411	0.291
mɑɣə'zɛin	mɑɣə'zɛintjə	tjə	81	81	1.0	0.988
mɑɣə'zɛin	mɑɣə'zɛinətjə	ətjə	308	95	0.308	0.220

The results thus consisted of one or multiple outputs per test-form, an example is shown in table 4. The first column is the stem in the phonological notation. The second column is the word with the learned suffix. The third column is the derived suffix. The fourth column is the scope. The fifth column contains the number of hits and the last two column contain reliability and confidence.

In the example above there are three possible guesses by the learner. For calculating the accuracy of the learner the case with the best reliability and best confidence is taken. For each item in the 3869 test items an output is obtained and compared with the CELEX data.

The results in total and per diminutive form are shown in table 5.

4.3. Comparative results of MGL with C4.5

The results obtained from the MGL were compared to those of Daelemans *et al*. This is shown in table 5, which shows the results obtained by choosing the MGL's top answers, either by confidence or by reliability, and the C4.5 results. Both the total results of all CELEX-input forms are shown, and per diminutive ending.

[1] In order the words mean: 'warehouse', 'sea voyage', 'artist's studio' and 'weaver'.

Table 5. Results of the MGL and C4.5.

Suffix	MGL-Confidence (% correct)	MGL-Reliability (% correct)	C4.5 (% correct)
Total:	**96.1**	**96.3**	**97.4**
jə	96.4	96.9	99.2
tjə	99.2	99.1	99.3
kjə	98.7	82.9	90.0
pjə	99.0	99.0	90.0
ətjə	78.1	81.9	84.0

The minimal generalization learner scores on 3869 items better when the best results of the minimal generalization learner are picked on reliability with a total score of 96.6 percent then on confidence with a total score of 96.4 percent. Daelemans *et al.* mention a total score of 97 percent on 3950 words. The confidence scores are the best on the suffix [kjə]. With 76 occurrences in the CELEX database the [kjə] is the rarest suffix, which indicates that the Minimal Generalization Learner scores good on irregularities. The rare [pjə] with 102 counts in the database scores better than the 90 percent outcome from the C4.5 learner on that suffix, but the [ətjə] with 370 items in the database scores slightly better with the C4.5 learner. In overall the C4.5 learner has better success with the most frequent items while the Minimal Generalization Learner scores best on the suffixes with lower frequency in the database.

4.4. Comparative results of MGL with behavioral data

A regression analysis was performed in order to assess the correlation between the Mean Behavioral Ratings and the Confidence Ratings produced by the MGL for each wug word diminutive form. Some of the diminutive forms were excluded from the analysis because the MGL algorithm did not produce a confidence rating for them. The forms that did not produce a rating trough the MGL were excluded from the analysis and the analysis was made for the remaining 91 forms. The obtained Pearson Correlation is 0.647 ($p<0.001$) and the R^2 corresponding to the regression line is 0.323 (figure 1). This indicates that an increase in the participants' mean survey ratings correspond to an increase of the confidence ratings of the MGL.

Finally, the regression analysis showed that only about 30% of the variability of the data obtained from the questionnaire can be explained by the confidence ratings of the MGL. Several explanations for this effect can be pointed out. First of all, as can be seen in figure 1, the MGL confidence ratings have a discrete distribution in the lowest levels and the behavioral data does not. This fact can be due to the calculation of the reliability rating by the MGL and the formation of "islands of reliability" (Albright & Hayes, 2003) that are very narrow (rules that can be applied in only a few cases, but with high confidence) and in such a way produce the same coefficient in a multitude of cases.

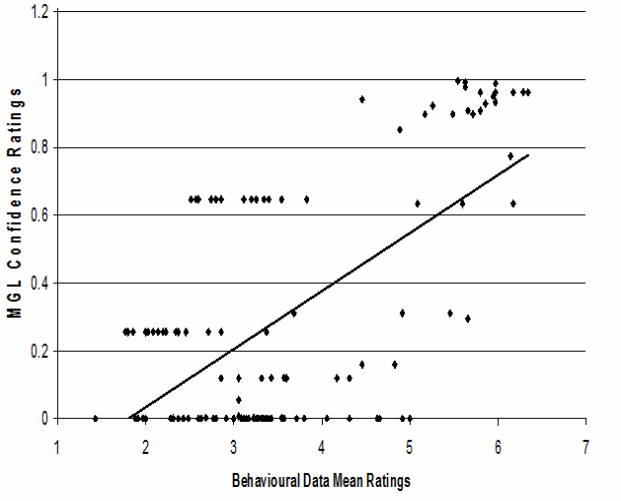

Figure 1. Linear regression between the mean behavior ratings for 91 wug-words from the questionnaire and the corresponding confidence ratings produced by the MGL. R^2=0.323 (t=7.69; p=0.00).

5. Discussion

The generalization model performed with high accuracy when choosing the diminutive suffixes for noun lemmas in CELEX. Its performance was as high as Daelemans *et al.*'s model, which shows that *a priori* knowledge of categories is not necessary for learning Dutch diminutives.

A more interesting test was to see whether the generalized rules found simulate human behaviour. Exposing people to unfamiliar words, wug words, will avoid instance-based answers and force the subjects to use generalized rules on the basis of phonotactics. MGL, if using the same rules, should then produce answers that are similar to the behavioural results. This was shown to be true for most of the diminutive categories, barring one. For 21 out of 25 wug words, the suffix with the highest MGL confidence corresponded to the answer with the highest score given by the subjects, or one of the not significantly different highest answers. Notably, the model did not generate any answers for wug words in the diminutive category [kjə], although it did generalize rules from CELEX with this suffix. Despite this obvious omission, the correlation found between human scores and MGL's confidence ratings was strong and significant. This could suggest that the underlying generalized rules are the same for people and the MGL.

The MGL lacked any answers with the suffix [kjə]. There is a noticeable problem when the wug word's lemma ended in [ŋ] without a preceding [I], but with a different vowel. The CELEX pairs used for training only contained nouns with the diminutive suffix [kjə] if the lemma ended on [Iŋ], whereas the survey contained several wug words

ending in [ŋ] with different preceding vowels. Interestingly people did give [kjə] answers with these wug words, where MGL did not. Two explanations are possible.

People were asked to rate each possible diminutive form for naturalness, or how 'Dutch' the word sounded. To obtain complete insight into people's preferences, subjects were not explicitly asked to generate suffixes themselves on the basis of a lemma, but to score all diminutive forms. With all possibilities written out in full, they were directly exposed to the, possibly incorrect, diminutive forms with the suffix [kjə] and lemmas ending in a [ŋ] preceded by one of these vowels: [øː], [ɑ]. [ɛ] or [ʏ]. Dutch does have diminutive words that end in [ŋkjə], and that have the just mentioned preceding vowels (such as *bankje* ('small sofa')), but these words contain the [k] in their lemma, making the suffix [jə] and not [kjə]. This could explain why people give diminutive forms such as *augengkje* a high score, because they sound natural if the lemma were to contain the [k], such as *augengk*. The MGL could not copy such behavior, for it can only generate answers on the basis of the actual lemma. If the MGL would be asked how natural the diminutive form is as a Dutch native word, it could possibly answer exactly like humans, because these wug words with [kjə] as a suffix do seem to comply to Dutch phonotactic rules. This could be tested by inputting the diminutive forms of the wug words in a MGL-model that can work with phonotactical constraints, such as the MGL-model implemented by Bart Cramer (unpublished manuscript). Another way to test this scenario is by giving human subjects the wug words such as *augengkje* and ask them to name the lemma. If people were to give *augenk* as a lemma, it would strongly suggest that people are indeed generating [kjə]-suffixes on the basis of phonotactics and not with the correct lemma.

The second explanation deals with the available information with which to learn the rules for Dutch diminutive forms. People could be choosing the suffix [kjə] on the basis of a rule that the MGL had not learned. The question then is whether the CELEX data is sufficient for the extraction of such a rule or that human subjects are able to generate the diminutive form in question without the need for positive examples. CELEX contains a great many Dutch words, but might exclude important categories such as proper names, verbs, foreign words or adjectives, of which many can be inflected into a diminutive form. If people are using such extra sources for learning the rule that causes the discussed behavior, then including them as training data for the MGL should allow the learner to come up with the same rule. If, however, extra sources for learning do not lead to similar behavior, it is possible to argue that the 'strategy' used by humans is different. The most extreme speculation would be that people are able to produce forms for which positive examples are not included in the learning material, because they can generalize more than the MGL.

With the exception of this special case, in which the answers between MGL and people differ significantly, the model seems to behave very similar to the natural system. Without any *a priori* categories assigned to the diminutive suffixes, it is clear that minimal generalization, as defined by Albright and Hayes (2002, 2003), can solve the well-defined language problem of Dutch diminutive forms.

Comparing the model's multiple answers per wug word and those of the subjects, shows a strong correlation. This can be explained very well if humans do not have a dual mechanism for regular and irregular cases (one of the assumptions on which MGL is based, see Albright & Hayes, 2003). Instead, humans might generate answers with multiple rules and pick the answer with the highest likeliness of being correct. The possible existence of more regular and irregular rules are explained by the model in having 'isles of reliability'. These are rules that have such a high scope and score (confidence) that they dominate all behavior within their scope, but leave room for specific exceptions. This would result in behavior that shows (seemingly) fixed rules dealing with regular cases, but it would allow for secondary answers with a lower confidence score, that are similar to people's secondary answers.

The previously described future experiments will determine what caused the observed omission of [kjə]-suffixes by the model. This excluded, the MGL as a model for human language learning is successful in learning rules for generating Dutch diminutive forms on the basis of naturally occurring training data.

Acknowledgments

First and foremost we would like to thank Adam Albright for supplying us with his (and Hayes') Java code for the MGL. He also gave valuable advice during his visit to the Netherlands.

Furthermore, we would like to thank Bart Cramer for introducing us to the theory of Minimal Generalization and allowing us insight into his implementation of the model.

Special thanks go to Elske van der Vaart for some final vital scrutinizing, something which comes natural to her.

Finally, we thank all friends and co-students who were bored and kind enough to help us with the online survey.

References

Albright, A. & Hayes, B. (2002). Modeling English Past Tense Intuitions with Minimal Generalization. in: *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*.

Albright, A. & Hayes, B. (2003). Rules vs. Analogy in English Past Tenses: A Computational/Experimental Study, *Cognition 90*, pp. 119-161.

Daelemans, W., Bereck, P. & Gillis, S. (1997). Data Mining as a Method for Linguistic Analysis: Dutch Diminutives, *Folia Linguistica*, *XXXI/I -2*, pp. 57-75.

Chomsky, N. (1965). *Aspects of theTheory of Syntax*. Cambridge, Mass.: MIT Press.

Chomsky, N. (1980). *Rules & Representation*. Cambridge, Mass.: MIT Press.
Crain, S. & Pietroski, P. (2001). Nature, Nurture and Universal Grammar. *Linguistics and Philosophy, 24*, pp. 139-186.
Cramer, B. (unpublished manuscript). *Stochastic rule-based phonotactics.* http://www.ai.rug.nl/~bcramer/publications/MSc_minor_thesis_Cramer.pdf
Duyck, W., Desmet, T., Verbeke, L. & Brysbaert, M. (2004). WordGen: A Tool for Word Selection and Non-Word Generation in Dutch, German, English, and French. *Behavior Research Methods, Instruments & Computers, 36(3),* pp. 488-499.
Mikheev, A. (1997). Automatic Rule Induction for Unknown-Word Guessing. *Computational Linguistics, 23*, 405-423.
Mintz, T. H., Newport, E. L., & Bever, T. G. (1995). Distributional Regularities of Form Class in Speech to Young Children. In Jill Beckman (Ed.), *Proceedings of the 25th Annual Meeting of the North Eastern Linguistics Society*. Amherst, Mass: GLSA.Mintz, Newport and Beer.
Pullum, Geoffrey K. & Barbara C. Scholz. (2002). Empirical Assessment of Stimulus Poverty Arguments. *The Linguistic Review 19* (special issue, nos. 1-2: 'A Review of "The Poverty of Stimulus Argument",' edited by Nancy Ritter), pp. 9-50.
Quinlan, J.R. (1987). *Programs for Machine Learning*. San Mateo: Morgan Kaufmann.
Quinlan, J.R. & Rivest, R.L. (1989). Inferring Decision Trees Using the Minimum Description Length Principle. *Information and Computation 80(3)*, pp. 227-248.
Trommelen, M. (1983). *The Syllable in Dutch*. Dordrecht: Foris.
Yang, C., (2004). Universal Grammar, Statistics or Both? *Trends in Cognitive Sciences, 8(10),* pp. 451-456.

Appendix A.

Rules of C4.5 for constructing nodes, and pruning.
- T is the set of training examples.
- A class is one of the possible outcome-categories.
- Features are the attributes of the input (such as phonetic features for language input).

Make-Decision-Tree (T) :

- If T contains cases belonging to class Cj, then the decision Tree for T is a leaf identifying class Cj.
- If T contains no cases. T is then a leaf. The overall majority class in the parent nodes of T is chosen as the identifying class for T.
- If T contains cases that belong to a mixture of classes. Then tests are constructed on single features. A test results in several subsets from the examples in T. The test with the highest Information Gain (based on entropy) will be used to construct the decision node for T. All constructed subsets will be input for Make-Decision-Tree (T).

Pruning (T) :

- Convert the paths from root to leaf node into rules. Example: If (Atr1 = X) and (Atr2 =Y), then outcome Category-1.
- Remove preconditions, if this would result in improving the estimated accuracy.
- Sort the rules per single class into subsets of rules.
- Sort the subsets on number of training cases covered by the subset.
- Sort the rules in the subsets based on their estimated accuracy (by calculating the Minimum Description Length over rules per class).
- Create a default rule, for the case none of the rules can be applied on the input.

Appendix B.

Below is shown the web-survey used for the behavioral study. The Dutch instruction reads, in short: "We are conducting a study in language acquisition of children, with the help of a computermodel. The following questions will be about wug-words, non-existing words, and we would like you to rate each possible form on a scale of 1 (not natural) till 7 (very natural). These answers will be compared to those of the model."

Hoe leren wij taal?

Wij doen een klein onderzoekje over hoe kinderen taal leren, daarvoor testen wij een computermodel. Leuk dat je mee wilt doen!

Hieronder volgt een lijst met WugWoorden, dit zijn niet bestaande woorden die Nederlands klinken. Voor elk van deze woorden tonen wij de 5 mogelijke verkleinwoorden die je er mee kunt maken.

Met deze vragenlijst willen we je vragen per mogelijke uitkomst aan te geven hoe goed je deze vindt. De uitkomsten zullen wij vergelijken met het computermodel.

Zou je eerst wat informatie over jezelf willen invullen? (doorhalen indien niet van toepassing)

Leeftijd: _____

Geslacht: M / V

Studie: Bachelorstudent faculteit Letteren / Masterstudent faculteit Letteren / werkende

Iets anders: _____

De woorden worden in willekeurige volgorden met de mogelijke uitkomsten weergegeven. Per woord willen wij vragen hoe natuurlijk een woord, op een schaal van 1 tot 7, je in de oren klinkt (zeg het woord ook even hardop, of in je hoofd). Per woord is ook aangegeven welke lettergreep de klemtoon heeft, deze lettergreep is ***dik en cursief*** afgedrukt.

De vragenlijst:

Score: 1 is heel onnatuurlijk tot en met 7 voor heel natuurlijk.

		1	2	3	4	5	6	7
1. bjarn	bjarnpje	☐	☐	☐	☐	☐	☐	☐
	bjarnje	☐	☐	☐	☐	☐	☐	☐
	bjarnkje	☐	☐	☐	☐	☐	☐	☐
	bjarnetje	☐	☐	☐	☐	☐	☐	☐
	bjarntje	☐	☐	☐	☐	☐	☐	☐
		1	2	3	4	5	6	7
2. kwolm	kwolmetje	☐	☐	☐	☐	☐	☐	☐
	kwolmpje	☐	☐	☐	☐	☐	☐	☐
	kwolmkje	☐	☐	☐	☐	☐	☐	☐
	kwolmje	☐	☐	☐	☐	☐	☐	☐
	kwolmtje	☐	☐	☐	☐	☐	☐	☐

Appendix C.

Feature file used for determining which feature each type of DISC phoneme has.

ASCII	DISC	syllabic	stressed	long	consonantal	sonorant	continuant	delayedrelease	approximant	tap	trill	nasal	voice	spreadgl	constrgl	LABIAL	round	labiodental	CORONAL	anterior
41	)	1	-1	1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
42	*	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	0
64	@	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
95	_	-1	-1	-1	1	-1	-1	1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	-1
124	|	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	0
125	}	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	0
60	<	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	0
65	A	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
97	a	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
98	b	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	1	-1	-1	-1	0
100	d	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	1
101	e	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
69	E	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
102	f	-1	-1	-1	1	-1	1	1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	-1	0
71	G	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
103	g	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
104	h	-1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	0
105	i	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
73	I	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
106	j	-1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
107	k	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
75	K	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	0
108	l	-1	-1	-1	1	1	1	0	1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	1
76	L	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	0
109	m	-1	-1	-1	1	1	-1	0	-1	-1	-1	1	1	-1	-1	1	-1	-1	-1	0
77	M	1	-1	-1	-1	1	1	0	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	0