RuG Rijksuniversiteit Groningen
Informatiekunde

bcn

Minimal generalization of Dutch diminutives

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

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

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

Final paper for master course Language Learning, february 2007

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 with 29 participants 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. This leads the authors to believe that the model's method of learning the necessary rules for this task displays characteristic similarities to the way humans learn these rules.

Word count 4587 (excluding abstract, tables, lists, figures, headings, references and appendices)

1. Introduction

One of the bigger debates on language acquisition is the question whether humans have innate knowledge for learning a language. 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. 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. When 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 build 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. Only a single study focusing on the same problem using machine learning has been discovered in the literature, which 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 preferred to the alternative. A parallel survey-study provides data about the native speaker's behavior in using Dutch diminutives, which enables 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 of the paper.

1.1. Dutch diminutives

A Dutch diminutive is the inflected form of a noun (other grammatical categories are possible too, such as adjectives, but these will be ignored for the sake of brevity), changing the meaning of the uninflected word to usually make it smaller, for example 'tafel' (table) becomes 'tafeltje' (little table) (for more reading, see Trommelen, 1983).

The standard rule for making Dutch diminutives is adding the ‘tje’ to the base of a noun. In general there are 5 known suffixes for the Dutch diminutive form, these are ‘tje’, ‘je’, ‘pje’, ‘kje’, and ‘etje’.

The frequency distribution in the CELEX database (Max Planck institute, Nijmegen) is shown in table 1, transcribed in DISC, a system developed by the same institute, which will be used throughout the paper. These results are based on 3889 unique diminutive nouns.

Table 1. Frequency per diminutive form.

tj@	1879	48.3
j@	1452	37.3
pj@	102	2.6
kj@	76	1.9
@tj@	370	9.5
Exceptions	10	0.2

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

- ‘j@’ is used after an obstruent like ‘pOpj@’.
- ‘pj@’ is used after a long vowel, diphthong or schwa, followed by /m/, like in 'bez@mpj@'. ‘pj@’ is also used after a short vowel followed by a liquid (/r/ or /l/) plus /m/.
- ‘@tj@’ is used after a nasal (/m/, /n/ or /N/) or the liquid /l/ preceded by a short vowel (as in: ‘romAn-@tj@’, ‘bAl-@tj@’). This diminutive ending is also added after monosyllabic words with a final /r/ that is preceded by a short vowel like in (‘bAr’).
- ‘kj@’ is used in multisyllabic words ending in /IN/ (like ‘konIN’) if the stress is on the penultimate syllable, like in 'sOlderIN-kj@' . The rule is strongly competing with the rule for '@tj@', for example in words like ‘lerlIN’ en ‘twelIN’, which are both ending on ‘@tj@’.
- ‘tj@’ is the default rule if any of the above is not applied.

Some words can have more than one suffix. See Daelemans et al. (1997) for a more detailed discussion on this on this topic, for this paper only some rules are discussed. 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/; ‘tj@’ and ‘@tj@’ are both possible.
- If a word ends on /m/; ‘pj@’ and ‘@tj@’ are both possible.
- If a word ends on /N/; ‘kj@’ and ‘@tj@’ are both possible.

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

- If a word ends on /p/ /b/ or /G/, ‘@tj@’ and ‘tj@’ are both possible.
- If a word ends in a long vowel followed by a sonorant after /m/, ‘tj@’ and ‘@tj@’ 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 algorithms from machine learning can be found in the area of Language Learning. A variety of methods on either natural languages or artificial ones, with an initial bias or without and on the basis of human learning methods or not. The latter of these decisions is not an easy one. Human inspired methods benefit from plausibility, but it is far from clear how humans learn on global or linguistic levels. The previously described C4.5 method is clearly not the way humans learn a language. It requires, for instance, predefined classes, which is an unlikely inborn attribute for humans to have. Even if this

likelihood is arguable, the enormous overhead requirements make the algorithm most unlikely. Storing all possible nodes and trying out numerous new nodes on the basis of high entropy for each and every new input is 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, inspired by presumed human intuitive and 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 some abstract classification. Secondly, if appropriate, the model should have multiple guesses for an answer, such as humans sometimes have more guesses as well, and the model should give an indication of how likely each answer is. The third criterion is the possibility for a model to distinguish detailed variations in language patterns. When generalizing rules (this will be explained shortly) the model should note small irregularities and learn these as well.

The developers of the minimal generalization learner deem these criteria necessary in order to begin mimicking the way a human learns generalization rules. The advantages above an algorithm such as C4.5 are not just the way in which it mimics humans better. For starters, the minimal generalization learner (MGL) has a constant pruning capacity. Furthermore, the multiple guesses the model can make actually keep the model flexible and capable of adjusting preferred generalizations whenever necessary. 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 input, a lemma and its derived form, both of them encoded in the corresponding phonetic form. For each of the phonemes the model needs to know what its phonetic or other audible features are, the same way a human can hear distinct differences between phonemes (presumably). These features enable the model to choose which regularities are 'audible' between the pairs, on the basis of which the model can then start to form generalizations. Here is an example for the Dutch words 'roos' and 'kaas'.

(1)
$$\begin{array}{llll} \emptyset \longrightarrow j@/\#r & o & s\_\# \\ \emptyset \longrightarrow j@/\#k & a & s\_\# \end{array}$$

These rules (1) indicate that the model learned to put 'je' (in DISC: 'j@') behind the word in order to get the correct diminutive form.

This simple step enables the model to learn for each pair the minimal change between the lemma and its derivative. However, it would be pointless to remember this for all input pairs. The model is therefore equipped with further generalization capacities (2).

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

This rule is a generalization of the two earlier mentioned word-based rules. This rule will add 'j@' after a word when it starts with a generic X (any set of phonemes), followed by a phoneme that has the minimal shared features of 'o' and 'a' (not all shared features are listed) and finally the 's'.

With multiple pairs of input, the model is capable of searching for regularities between pairs. It creates a new rule that generalizes with a minimum of feature-changes. It is therefore called minimum generalization, and this kind of rule-forming is the brunt of the model. With each new input, it searches whether it fits an already made rule, and if not, it adapts the most appropriate rule to 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 better, since the new input-form could be an exception. All rules need to defend their usefulness by remembering their hits, the number of times the rule created 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, all applicable rules compete to generate output.

The model is now capable of constantly generating generalizations on the basis of phonetic features, trying to maximize correct generalizations by penalizing rules which either generate incorrect output forms or do not generate output at all.

One of the elegant behaviors of this model is its indiscriminate attitude towards regular and irregular forms. The model quite simply does not care and will generate generalizations wherever it can. Regular forms will naturally lead to rules with larger scope, but irregular forms still have large similarities and in their most exceptional case will still lead to a rule with a scope of one. Albright and Hayes claim that humans learn in a similar fashion, not fundamentally discriminating between regular and irregular forms in the way they are learned.

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 the model of choice for application on 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 provides an equal rate of correct answers as the C4.5 which will indicate 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. For this test a number of wug-words were formulated in a way that every diminutive form in the Dutch language can be applied at least to some of the wug-words. Wug-words are invented words that do not exist in the natural language, but that follow the phonetic rules of the language. With these words we could test which diminutive form participants would prefer. Also, we wanted to obtain a coefficient of confidence for every possible output form in order to compare the sequence of most confident to less confident with the one produced by the Minimal Generalization Learner.

3.1. Method

Wug-words were created using a program which on the basis of the CELEX corpora and the phonotactic rules of the language creates 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 34 were selected for the study (see table 2 for a list) on the basis of similarity to real Dutch words, while making sure every diminutive ending had certain candidates. The questionnaire consisted of 170 questions corresponding to al five possible forms of each wug-word. Participants were required to give their confidence rating (How well does this form sounds to you?) on a scale from 1 to 7. The questionnaire was uploaded on the web and participants responded on-line. (for an example of the questionnaire see Appendix B)

3.2. Participants

29 students from the University of Groningen participated in the study.

3.3. Results and Discussion

The ratings were analyzed in term of the discriminability between the diminutive forms. For each wug-word a five level ANOVA was computed (nonparametric Kruscal-Wallis test for the words that did not cover the assumption of equality of variances) 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 most of the wug-words, although for some of the words it is not possible to identify a single diminutive form that is preferred by the participants. The last fact can be explained by the nature of the questionnaire that was used. For each option only the graphical transcription was presented without the stress. The stress assumed by different participants can be different, which could lead to different choices (see *5.2.*).

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 eventually 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.

Table 2. A list of all 34 wug-words and the behavioral survey results. The right column shows the order of preference, where a comma indicates a non-significant difference and a '>' indicates a significant preference.

Wug-word		Ordering of the diminutive forms
Ambing	(F(4,165) =100.48; p =0.000)	Ambingkje, Ambingetje > Ambingtje, Ambingje, Ambingpje
Belui	(X^2(4) = 87.45; p =0.000)	Beluitje, Beluikje, Beluipje, Beluietje, Beluije
Benre	(F(4,165) =16.07; p =0.000)	Benretje > Benreetje, Benrekje, Benrepje > Benreje
Bexeid	(X^2(4) = 91.2; p =0.000)	Bexeidje, Bexeidetje, Bexeidtje, Bexeidkje, Bexeidpje
Bulkan	(X^2(4) =82.18; p =0.000)	Bulkanetje, Bulkantje, Bulkanje, Bulkankje, Bulkanpje
Deron	(F(4,165) =47.18; p =0.000)	Deronetje, Derontje > Deronkje > Deronje, Deronpje
Dofman	(X^2(4) =88.68; p =0.000)	Dofmanetje, Dofmantje, Dofmankje, Dofmanje, Dofmanpje
Egkel	(F(4,165) =23.82; p =0.000)	Egkeltje > Egkeletje, Egkelpje, Egkelkje > Egkelje
Etken	(F(4,165) =41.41; p =0.000)	Etkentje, Etkenetje > Etkenkje, Etkenpje, Etkenje
Euk	(X^2(4) =93.77; p =0.000)	Eukje, Euketje, Euktje, Eukkje, Eukpje
Fuk	(X^2(4) =98.33; p =0.000)	Fukje, Fuketje, Fuktje, Fukkje, Fukpje
Gug	(X^2(4) = 90.45; p =0.000)	Gugetje, Gugje, Gugtje, Gugpje, Gugkje
Hir	(F(4,165) =13.53; p =0.000)	Hirtje, Hiretje, Hirkje, Hirpje > Hirje
Jub	(F(4,165) = 48.78; p =0.000)	Jubetje, Jubje > Jubtje, Jubkje, Jubpje
Kaptel	(F(4,165) = 39.47; p =0.000)	Kapteletje > Kapteltje > Kaptelkje, Kaptelpje, Kaptelje
Knagem	(F(4,165) = 40.94; p =0.000)	Knagempje > Knagemetje, Knagemkje, Knagemtje, Knagemje
Kolel	(F(4,165) = 27.28; p =0.000)	Koleletje, Koleltje > Kolelkje, Kolelpje, Kolelje
Kulia	(X^2(4) = 96.18; p =0.000)	Kuliatje, Kuliakje, Kuliapje, Kuliaje, Kuliaetje
Matia	(X^2(4) = 83.76; p =0.000)	Matiatje, Matiapje, Matiakje, Matiaje, Matiaetje
Nalten	(F(4,165) = 28.07; p =0.000)	Naltentje, Naltenetje > Naltenkje > Naltenpje, Naltenje
Pakden	(F(4,165) = 71.45; p =0.000)	Pakdenetje, Pakdentje > Pakdenkje > Pakdenje, Pakdenpje
Paping	(F(4,165) = 61.22; p =0.000)	Papingkje, Papingetje > Papingtje, Papingje, Papingpje
Pinnem	(F(4,165) = 20.33; p =0.000)	Pinnempje > Pinnemetje, Pinnemtje, Pinnemkje, Pinnemje
Qum	(X^2(4) =97.22; p =0.000)	Qumetje, Qumpje, Qumtje, Qumkje, Qumje
Redek	(X^2(4) =103.92; p =0.000)	Redekje, Redeketje, Redektje, Redekkje, Redekpje
Rewin	(F(4,165) =34.45; p =0.000)	Rewintje, Rewinetje > Rewinkje > Rewinje, Rewinpje
Sfart	(F(4,165) =57.24; p =0.000)	Sfartje > Sfartetje > Sfarttje, Sfartkje, Sfartpje
Sleub	(F(4,165) =33.23; p =0.000)	Sleubje > Sleubetje > Sleubtje, Sleubkje, Sleubpje
Ult	(X^2(4) =83.56; p =0.000)	Ultje, Ultetje, Ulttje, Ultkje, Ultpje
Urs	(F(4,165) =17.96; p =0.000)	Ursje > Ursetje, Urstje, Urskje, Urspje
Vrote	(X^2(4) =51.78; p =0.000)	Vrotetje, Vroteetje, Vrotekje, Vrotepje, Vroteje
Wur	(F(4,165) =28.2; p =0.000)	Wurtje, Wuretje, Wurkje, Wurpje > Wurje
Xezen	(X^2(4) =64.25; p = 0.000)	Xezentje, Xezenetje, Xezenkje, Xezenpje, Xezenje

4.2. Testing the MGL with CELEX-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.

4.2.1. Introduction on CELEX

CELEX was founded in Nijmegen in 1986 under supervision of several Dutch-based research centers, most notably the Max Planck institute 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

DISC	DISC + Dim.	Freq.	Stem	Stem + Dim.
mAGa'zKn	mAGa'zKntj@	1	magazijn	magazijntje
'zerKs	'zerKsj@	4	zeereis	zeereisje
At@l'je	At@l'jetj@	2	atelier	ateliertje
wev@'rK	wev@'rKtj@	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 DISC notation. The second column is the Dutch diminutive form in DISC notation. 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 notation and primarily used as annotation of the data.

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 'magazijn'.

Input	Output	Dim.	Scope	Hits	Rel.	Conf.
mAGa'zKn	mAGa'zKnj@	j@	3038	1250	0.411	0.291
mAGa'zKn	mAGa'zKntj@	tj@	81	81	1.0	0.988
mAGa'zKn	mAGa'zKn@tj@	@tj@	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 DISC 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 result obtained from the MGL were compared to those of Daelemans. 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.

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**
j@	96.4	96.9	99.2
tj@	99.2	99.1	99.3
kj@	98.7	82.9	90.0
pj@	99.0	99.0	90.0
@tj@	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 mentions a total score of 97 percent on 3950 words. The confidence scores are the best on the suffix 'kj@'. With 76 occurrences in the CELEX database the 'kj@' is the rarest suffix, which indicates that the Minimal Generalization Learner scores good on irregularities. The rare 'pj@' with 102 counts in the database scores better than the 90 percent outcome from the C4.5 learner on that suffix, but the '@tj@' with 370 items in the database scores slightly better with the C4.5 learner. In overall the C4.5 learner has a better success on the most items where the Minimal Generalization Learner scores best on the suffixes with inferior frequency in the database.

4.4. Comparative results of MGL with behavioral data

A regression analysis was performed in order to access the correlation between the Mean Behavioral Ratings and the Confidence Ratings produced by the MGL for each

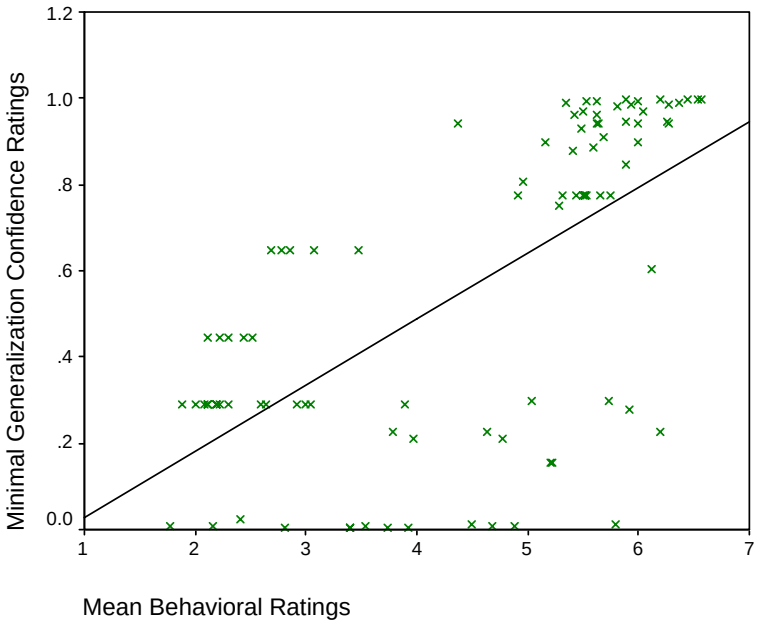

Figure 1. Linear regression between the mean behavior ratings for 91 wug-words from the questionnaire and the corresponding confidence ratings produced by the MGL. R^2=0.418 (β=0.64; t=8.00; p=0.00).

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=0.000) and the R^2 corresponding to the regression line is 0.418 (figure 1). This indicates that an increase in the participants' mean survey ratings correspond to an increase of the confidence ratings of the MGL.

5. Discussion

5.1. Comparative analysis

The main aim of the study was to discover whether the MGL was able to mimic human learning behavior in the case of Dutch diminutive forming, and as a special case performs as well as the C4.5 model without its need for *a priori* categorical knowledge.

To test whether the MGL model was capable of learning the problem at all, a ten-fold cross validation with all 3869 input items was used. This resulted in a score of about 96% of correct predictions (96.1 for the confidence and 96.3 for the reliability ratings), which indicates that the model is suitable for the chosen problem.

Furthermore, it was argued that because no *a priori* knowledge about categories was necessary to learn the Dutch diminutive forming, the MGL model would perform at least as well as the C4.5 model. This is confirmed by the results, which show no real differences in performance between the models. The MGL is slightly outperformed when all inputs are considered. C4.5, however, has problems with the more exceptional cases of 'pj@' and 'kj@', where the MGL does not. This is likely due to the MGL's capability of dealing with irregular forms, thanks to its sensitiveness to less common generalizations, and the rigidness of C4.5 caused by the *a priori* introduction of the categories. What is most important, is the fact that MGL certainly does not perform worse than C4.5, indicating that knowledge about categories is not an essential prerequisite for learning the rules of forming Dutch diminutives.

The model was able to generate multiple answers for each wug-word, enabling comparison with participants asked to rate each diminutive ending per wug-word. There is a strong positive correlation between the answers given by the MGL and the participants. This indicates that the model seems to correctly mimic human behavior for previously unknown words. Humans do not give just one answer per novel word, but they give several of them with different confidence ratings on each of them. The model does something very similar, and therefore we might argue that the model captures the intuitive approach the participants use when they encountered words that do not exist.

5.2. Problems

Some limitations of the study can be pointed out. First of all, concerning the questionnaire, the wug-words and corresponding forms were presented in a written format without any stress added. As was already mentioned above, different participants can assume different stress for the same wug-word (or even the same participants can use different criteria per diminutive form of the same wug-word). The result is that for some of the wug-words no single most confident form can be identified.

Another point that is worth mentioning is that the MGL did not gave a rating for each possible diminutive form of each wug-word, as participants in the questionnaire did. Participants were implicitly forced (there was no non-answer option) to give a rating for each form, and it is not known if they would produce all of them if asked in an open ended question.

Concerning the diminutive forms found in CELEX, there appear to be a few irregular forms outside the common rule of adding one of the five suffixes. The most striking exception is entirely new suffix, namely 'k@', such as in 'mAn' → 'mAn@k@', but it is highly uncommon. Furthermore, phonetic changes of the lemma occur, like in 'sxIp' → 'sxepj@', or even a deletion of the /n/ at the end of a lemma, like 'jON@n' → 'jON@tj@'. The MGL deals with these irregulars by learning rules that add the new suffix, in these cases '@k@' and 'epj@'. In the case of 'jON@tj@' the learner will add a 'tj@' instead of '@tj@' to the lemma. In total only 10 cases of irregulars were found in the CELEX data. The irregulars do have a

small influence on the learner, by adding guesses of irregular suffix for many of the input-forms, but the confidence for such outputs is always very low due to a very low hit/scope ratio. Furthermore, we did not notice any effects on the wug-words test.

Finally, the regression analysis showed that only about 40% 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.

5.3. Improvements and further research

First of all, the behavioral data should be collected in a way that reflects the exact characteristics of the input used for the MGL by including the stress in the wug-words.

The model can be tested with different amounts of input in order to check how much is needed for the learning of each rule.

Greater exposure to natural input could be generated by making use of the flexible nature of Dutch diminutive forms, which are not only applicable to nouns (current model), but to names, adjectives (making it a noun) and adverbs as well. Increasing the training-set, especially for the irregular forms, could increase the model's learning capacity further.

To test further intuitive notions of human learning, the test data could be extended to non-Dutch words. Similar to wug-words, both human and MGL results can be compared to test the model's mimicking capacity for human intuitive behavior to novel input.

Acknowledgments

First and foremost we would like to thank Adam Albright for supplying us with his (and Hayes's) 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.

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.

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.

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

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). Als je klaar bent met het invullen van de lijst, kan je op de knop 'klaar' drukken. (1 is heel onnatuurlijk tot/met 7 heel natuurlijk)

1.

		1	2	3	4	5	6	7
urs	ursje	○	◉	○	○	○	○	○
	urskje	○	○	○	○	○	◉	○
	ursetje	○	◉	○	○	○	○	○
	urspje	○	○	◉	○	○	○	○
	urstje	○	○	○	○	○	○	◉

2.

		1	2	3	4	5	6	7
ult	ultkje	◉	○	○	○	○	○	○
	ultje	○	○	○	○	◉	○	○
	ulttje	○	○	○	○	○	◉	○
	ultpje	○	○	○	○	○	○	◉
	ultetje	◉	○	○	○	○	○	○

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