Learning Rules and Categorization Networks for Language Standardization

Gerhard B van Huyssteen
Human Language Technology Group
Council for Scientific and Industrial Research
Pretoria, South Africa
gvhuyssteen@csir.co.za

Marelie H Davel
Human Language Technology Group
Council for Scientific and Industrial Research
Pretoria, South Africa
mdavel@csir.co.za

Abstract

In this research, we use machine learning techniques to provide solutions for descriptive linguists in the domain of language standardization. With regard to the personal name construction in Afrikaans, we perform function learning from word pairs using the Default&Refine algorithm. We demonstrate how the extracted rules can be used to identify irregularities in previously standardized constructions and to predict new forms of unseen words. In addition, we define a generic, automated process that allows us to extract constructional schemas and present these visually as categorization networks, similar to what is often being used in Cognitive Grammar. We conclude that computational modeling of constructions can contribute to new descriptive linguistic insights, and to practical language solutions.

1 Introduction

In the main, constructionist approaches to grammar focus on discovering generalizations in language by analyzing clusters of usage-based instances of linguistic phenomena. Similarly, computational linguistic approaches to grammar learning aim to discover these very same patterns, using automated techniques such as machine learning (ML).

In this research, we use techniques from ML to analyze and predict irregular phenomena with limited data available, and then represent these phenomena visually in a way that is compatible with the Cognitive Grammar descriptive framework (as a constructionist approach to grammar; henceforth CG). Our grand goal is to develop language technology tools that could be used in descriptive linguistics. Specifically, we aim to (1) develop a predictor that could suggest derivational forms for novel base-forms; and (2) automatically extract categorization networks (i.e. constructional schemas and the relationships between them) from a dataset, which could serve as a heuristic input to descriptive linguistics.

2 Contextualization

This research originates from a practical problem related to language standardization. Similar to standardization bodies for languages like Dutch, and German, the "Afrikaanse Taalkommisie" (TK) is the official body responsible for the description and regulation of Afrikaans spelling. The TK regularly publishes the official orthography of Afrikaans in the form of the *Afrikaanse Woordelys en Spelreëls* ('Afrikaans Wordlist and Spelling Rules'; AWS (Taalkommissie, 2009)).

One of the challenges faced by the TK is to standardize the spelling of foreign place names (including names of countries, cities, regions, provinces, etc.), and their derived forms (i.e. adjectives, such as *Amerika·ans* 'American'; and personal names, such as *Amerika·ner* 'person from America'). In the absence of sufficient usage-based

Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics, pages 39–46,
Los Angeles, California, June 2010.

evidence, many variant forms are often being accepted, either related to spelling or derivation; compare for instance the variant spelling forms *Maskat* or *Masqat* or *Muskat* 'Muscat', or the variant derivational forms *Turkmenistan·i* or *Turkmenistan·ner* 'person from Turkmenistan'. The TK is therefore challenged with the task to give guidelines regarding spelling and derivation, while faced with highly irregular and sparse data containing many variants.

We contribute to address this challenge by discovering the constructions in seemingly unsystematic and irregular data. Based on our tools and outputs, the TK could then revise existing irregularities and variants, or use these tools to guide future decisions.

3 Related Work

3.1 Constructional Schemas

Morphological constructions can be defined as composite symbolic assemblies (i.e. complex form-meaning pairings) smaller than phrases, consisting of component structures between which valence relations hold (Van Huyssteen, 2010; see also Tuggy, 2005). One of the main component structures in morphological constructions is the morpheme, which is simply defined as a simplex symbolic unit in the language system (i.e. it does not contain smaller symbolic units as subparts). More schematic symbolic assemblies (i.e. less specified in their characterization) are referred to as constructional schemas.

Constructional schemas can be represented as a network with relationships of categorization holding between different constructional schemas; these categorization networks provide the structural description of a construction (Langacker, 2008: 222). In the representations used in CG, categorization relationships of elaboration (i.e. full instantiations of a schema), extension (i.e. partial instantiations), and correspondence are specified. Entrenchment and ease of activation is indicated by the thickness of boxes: the thicker the line of a box, the more prototypical that unit is (Langacker, 2008: 226; see also Figure 5).

The aim of descriptive linguistics is to postulate categorization networks that describe a construction in a language, based on usage data. Our research contributes to this aim by automatically creating visual representations of such language models. For our current research, we are specifically interested in the personal name construction in Afrikaans.

3.2 Afrikaans Personal Name Construction

Formation of personal names by means of a personal name creating derivational suffix (NR_{PERS}) is a productive process in many languages. The specific category that we are investigating in this research is personal names derived from place names, such as *Trinidad·ees* 'person from Trinidad'.

In one of the standard works on derivation in Afrikaans, Kempen (1969) identifies a number of NR_{PERS} suffixes that are used in derivations from place names. He finds that there is no obvious systematicity in their distribution (based on a dataset of 132 instances), but concludes that, in derivations of foreign place names, the **-ees** and **-s** morphemes are most frequently used, with some distribution also over **-i**, **-n** (especially **-aan**) and **-r**. In addition to some of the morphemes mentioned by Kempen (1969), Combrink (1990) also mentions a few, while excluding others. In as far as we know, no other description of this construction in Afrikaans has been done, and based on the difference between Combrink (1990) and Kempen (1969), we can also deduct that there is no comprehensive understanding of this construction.

Personal names from place names can be formed in four basic ways in Afrikaans: (1) suffixation (*Aruba·an* 'Arubian'); (2) zero derivation (*Aberdeen* 'person from Aberdeen'); (3) clipping and back-formation (*Turk<Turkye* 'person from Turkey'; *Armeen<Armenië* 'person from Armenia'); and (4) lexicalization (*Cornwallis>Korniër* 'person from Cornwallis'). In a rather large number of cases (119 in our dataset of 1,034; see 5.1) none of the above strategies can be applied, and then paraphrasing is being used (e.g. *'n persoon van Akkra* 'a person from Accra').

Variants of morphemes (i.e. allomorphs) exist for phonological reasons, of which a linking element is the most prominent (Combrink, 1990). Compare for example **-aar** in *Brussel·aar* 'person from Brussels' (where the base-form is polysyllabic) vs. **-enaar** in *Delft·enaar* 'person from Delft' (where the base-form is monosyllabic; *Delftenaar* could therefore also be analyzed as *Delft·en·aar*).

For our purposes, we consider **-enaar** as an allomorph (i.e. elaboration) of –**aar**, and is identified as such in our categorization network (see Figure 5). Similarly, we classify morphemes as allomorphs in cases where an allomorph exists due to identical vowel deletion (e.g. **-an** as a variant of **-aan** when it combines with a base-form ending on an *-a*, as in *Afrika·an* 'person from Africa'), as well as consonant doubling after a short, stressed syllable in the auslaut (e.g. **-mer** as a variant of **-er**, as in *Amsterdam·mer* 'person from Amsterdam').

3.3 Automatic Extraction of Constructional Schemas

Computational modeling of morphology is a vast subfield in computational linguistics, gaining popularity since the 1980s. Pioneering work in the field has been done within the two-level morphology framework, and elaborations on this framework can be considered the basis of state-of-the-art morphological analyzers today. However, since constructing such analyzers manually is hugely expensive in terms of time and human effort, the approach does not scale well for new languages.

To overcome this obstacle, many computational linguists have developed techniques towards the automatic learning of morphology (e.g. Goldsmith, 2001). A key goal is to be able to produce a morphological analysis of the words of a corpus when only provided with the unannotated corpus.

We are interested in the related goal of function learning: given a base-form of a word, learn other forms of the word. Most typically, function learning takes pairs of words (base-forms plus inflected/derived forms) as input to discover patterns in the data. This is also the paradigm used in the current paper.

Several ML techniques have been used to solve specific function learning tasks (such as learning the past tense form of the English verb). Approaches include the use of decision trees, neural networks, inductive logic programming, and statistical approaches (Shalonova & Flach, 2007).

We are not aware of any work related to the automated learning of categorization networks specifically.

4 Approach

Our research has two complementary goals, dealt with separately: (1) to develop a predictor that can suggest potential derivational forms for novel base-forms (and alternative forms for existing base-forms with irregular forms); and (2) to automatically extract categorization networks that are easily interpretable by linguists.

4.1 Prediction of Derivational Forms

In order to analyze existing and predict new derivational forms, we use the Default&Refine (D&R) algorithm (Davel & Barnard, 2004). This algorithm extracts context-sensitive rules from discrete data, and is particularly effective when learning from small training sets. It has the additional advantage that rules generated are interpretable by humans. When applied to the grapheme-to-phoneme prediction task, it has been shown to outperform comparative algorithms (Davel & Barnard, 2008).

The D&R algorithm defines a set of templates and then uses a greedy search to find the most general rule (matching the templates) that describes the training data in question. Examples that are successfully explained by this rule are removed from the data set and the process repeated. Whenever a new rule contradicts examples previously dealt with successfully, these are again added to the training data to be "re-explained" by a later rule. The rule set therefore captures hierarchical default behavior: the last rule defines the default behavior for a specific pattern, and acts as a back-off rule to the second-last (more refined) rule, which would capture deviations from default behavior. The second-last rule would then act as back-off to the third-last rule, and so forth. Rules are therefore explicitly ordered according to the reverse rule extraction order. (The rule extracted first is matched last.)

Once a set of rules have been generated, these describe the training data completely. In addition, by tracing each of the possible rules that may apply to a new pattern (in order), various alternative derivational forms are identified, along with the evidence supporting each option (as in Table 2).

4.2 Extraction of Categorization Networks

While the D&R rules extracted in Section **Error! Reference source not found.** provide a perspective on the phenomena that occur, these rule sets could become extremely large and, accordingly, more difficult to interpret. We therefore attempt to extract categorization networks (*a la* CG) as visual

representations in a fully automated fashion. These networks are more easily interpretable, especially to humans.

An iterative string matching process is used to structure "potential morphemes" within a directed tree. Our main assumptions are that:

- the only input to the process consists of a set of unannotated word pairs: base-form + derivational form;
- a morpheme is added as a suffix;
- allomorphs are either shorter than the main morpheme (i.e. characters removed) or longer (i.e. characters added); and
- preference is given to larger strings that occur systematically in the training data.

The following steps are followed:

1. Generate a list of initial *transformation classes* based on the word pairs provided. These are derived through a comparison based on the longest common substring of the derivational form and its respective base-form (see Table 1). The classes specify the character string to be removed from the base-form (if any), and the replacement string; note that ellipses indicates the base-form (or part of it), and curly brackets indicate deletions (i.e. in *China*, delete the *-a*, and then add **-ees**). If a place name and its personal name are identical, the class will be "0".

Table 1: Examples of transformation classes

Place name	Personal name	Class (constructional schema)
Aberdeen	*Aberdeen*	[[x] [0]]
Amerika	*Amerikaner*	[[…] [ner]]
China	*Chinees*	[[…{a}] [ees]]

2. Create a list of all transformation classes and, per transformation class, a set of all derivational forms (referred to as the *transformation derivations set*).
3. For each transformation derivations set, find the largest end-of-word string common to all members of that set (the *set best string*). The set of all "set best strings" are referred to as the *best string list* and can be interpreted as a set of candidate morphemes.
4. For each transformation derivations set, consider the elements in the best string list, and determine if any subsets of the current set exist that match a larger string currently in the best string list. If so, *partition the set into subsets* accordingly. (Each subset is therefore identified by both a transformation class and a best string. For example, three different sets, each with a different best string may be related to a single transformation class. This makes it possible to identify situations where an allomorph is created in other ways than simply adding the morpheme as a suffix.)
5. For each subset, *update the set best string* based on the latest partition; update the best string list to reflect new best strings created.
6. *Repeat* steps (4) and (5) until no further changes are made. The set of morphemes are considered stable, and it now remains to structure these elements into a visual categorization network.
7. In order to create the categorization network, we start with an empty directed graph. For each set best string, create a list of all the transformation classes that are applicable (as calculated above) and *add these transformation classes from largest to smallest* to a single branch of the tree. (One branch is created for each string in the best string list, and is a first attempt at capturing a morpheme along with its different variations.)
8. Consider the nodes at each level (all nodes that have the same node as parent) and wherever one node fully contains another, *move the contained node* to become the parent of the other (cutting the link between the original parent node and the contained node). This process ensures that morpheme candidates that are actually variations of other morphemes are suppressed at each level of the tree.
9. Now *combine* any nodes that occur in different places in the tree but have *identical transformation classes*, by merging the lower node with the higher node. Only identical transformation classes are merged.
10. For each node in the final tree, consider whether the left hand side of the transformation class can be *refined*, specifically by adding additional matching characters based on the final transformation derivations set.

The result of this process is a set of final transformation classes, each describing a constructional schema, and the relationships among these constructional schemas, displayed as a categorization network.

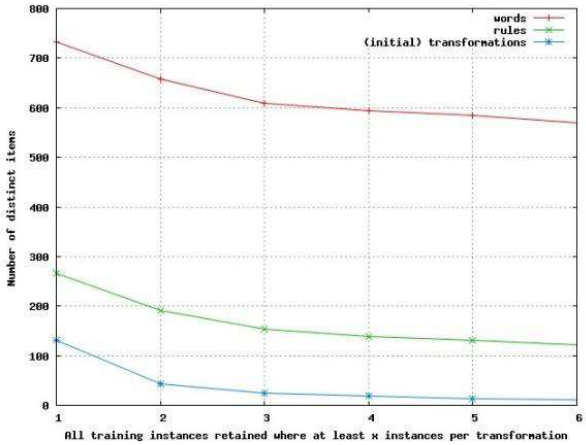

Figure 1: Number of words, rules and initial transformations for the various person-x data sets

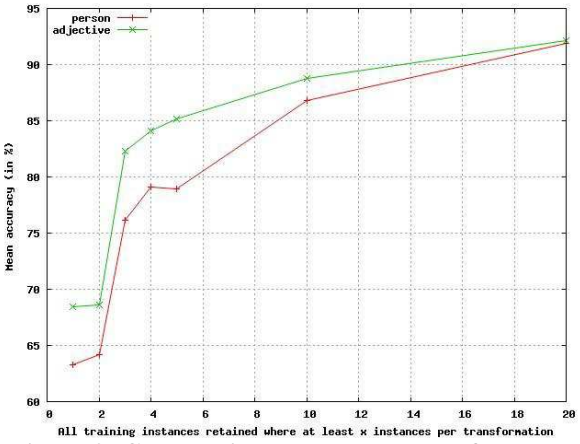

Figure 2: Cross-validated rule accuracy for the person-x and adjective-x data sets.

5 Experimental Setup and Results

5.1 Data

The dataset that we use is the list of foreign place names and their corresponding personal names from the AWS (Taalkommissie, 2009). For purposes of brevity, we only report on suffixation and back-formation, and exclude cases with variant morphemes, zero derivation and clipping, as well as all cases of paraphrasing. 732 instances are retained (from the original dataset of 1,034 instances).

A supplementary dataset consisting of adjectival derivations of place names was also taken from the AWS and treated in the same manner as the personal names; this dataset is used in Section 6.3 to verify certain of the findings. This set contains 786 instances.

5.2 Development of Predictor

The full dataset is highly irregular, containing many transformation classes that occur only once. We are interested in these irregularities (in order to identify words that may need further review), as well as in more systematic phenomena that occur in the data. We therefore create different data sets; in each set (referred to as *person-x*) we only retain those instances that occur *x* or more times in the transformations. (The *person-1* set therefore contains all training data, including all exceptions, while the person-6 set only contains transformations supported by 6 or more instances.) In Figure 1 the number of words and number of unique transformation classes are displayed for each *person-x* data set.

In order to verify the accuracy of our extracted rules, we use 10-fold cross-validation to obtain a mean accuracy per data set, as depicted in Figure 2 (labeled "*person*"). We also generate a rule set from the training and test data combined: this larger set is used to extract categorization networks.

When the rule set is structured as a graph (called a rule network), the data can be interpreted as follows: the root node indicates the default transformation, which applies unless any child node is matched by the base-form, which again only applies unless a child of the child node matches the base-form (and so forth), which indicates that a more refined rule should be applied. A small part of a rule network is displayed in Figure 3, with each node listing the end-of-word string of the base-form that will trigger the rule, the transformation rule that will be applied, and the number of instances of the rule in the training data. The complete rule network is very large: 266 nodes for the *person-1* data set, as indicated in Figure 1.

As was expected, a large number of exceptional rules are generated, indicating much inconsistency in how derivations are formed. For the *person-1* data set, 217 exceptions are identified. For each of these exceptions, alternatives are suggested in order of prototypicality by tracing the rule network, as illustrated for the base-form *Smirna* in Table 2. Automatically generated tables like these provide a practical tool for language standardization.

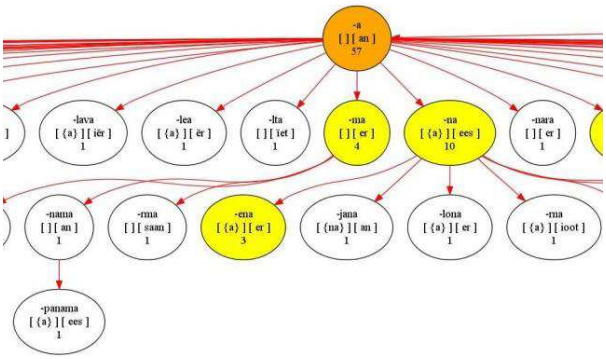

Figure 3: A small subsection of a rule network

Table 2: Alternative suggestions for the exception: *Smirna -> Smirnioot*

Alternative	Instances	Examples
Smirna	*1*	*Smirna>Smirnioot*
Smirnees	*1*	*Navarra>Navarrees*
Smirnaan	*58*	*Sparta>Spartaan* *Astana>Astanaan*
Smirnaer	*155*	*Hiroshima>Hiroshimaer* *Breda>Bredaer*

5.3 Development of Categorization Networks

The categorization network in Figure 5 was compiled automatically, as described in 4.2. Note that this specific categorization network is based on construction schemas with three or more supporting examples per node; for the sake of brevity, we do not include the full categorization network (based on all the examples) in this paper.

The relative prototypicality of constructional schemas (indicated by the thickness of lines in Figure 5) is determined post hoc by observing distribution frequencies. We obtain four natural clusters in this way: highly prototypical (hundred or more instantiations), prototypical (forty or more instantiations), less prototypical (three or more instantiations), and unprototypical (less than three instantiations, therefore also including exceptions); the latter category is not included in Figure 5.

Full instantiations of a schema (i.e. relationships of elaboration) is indicated with solid arrows; the highest node in our network represents the semantic pole, and is here simply indicated as [[PLACE X] [NR_{PERS}]]. For each node in the network, we also indicate the class frequency, and provide three examples of the base-form.

6 Discussion

6.1 Predictor

The extracted rules immediately provide us with:

- An indication of the predictability of the data (rule accuracy);
- A set of all exceptions (single instances that require an individual rule to describe that instance); and
- A predictor of new forms (applying the rules to unseen words).

From the accuracies depicted in Figure 2, it is clear that the full data set, including all phenomena that only occur once, describes a difficult learning task, with an overall accuracy of only 63.2% achieved. When more systematic phenomena are investigated (i.e. transformations with six or more instances), our classification accuracy quickly increases above 80%, indicating that the predictor is in fact usable. An error analysis reveals that improvements may be possible by taking pronunciation information into account (stress patterns, syllable information, consonant categories, etc.).

A standardization body such as the TK could use the automatically generated list of exceptions (similar to Table 2) to review prior standardization decisions. In addition, the predictor can be used to suggest derivational forms for novel base-forms, which could then be verified with usage data.

6.2 Categorization Networks

From Figure 5, observe that we have identified seven basic morphemes (i.e. nodes on the highest level), viz. **-aan**, **-aar**, **-ees**, **-er**, **-i**, **-iet** and **-ër**; with the exception of the latter, all these correspond to the morphemes identified by Kempen (1969) and Combrink (1990). Linguistically speaking, **-ër** is actually an extension of the [[…] [er]] construction, since the e-trema is used in Afrikaans orthography as a variant of the letter "e" to signify a syllable with a null onset, preceded by a syllable without a coda. However, our algorithm treated **-er** and **-ër** as two separate morphemes.

We can also observe that the [[…] [er]] constructional schema can be considered the most prototypical schema (based on frequency). Other prototypical constructional schemas include [[…a] [an]], [[…] [ner]] and [[…] [ër]] (with the latter two actually instantiations of [[…] [er]]). Within a

CG framework, it is assumed that these prototypical constructional schemas are more likely to be activated for the categorization of novel examples.

This observation contradicts Kempen's (1969) finding that there is no obvious systematicity in the distribution of personal name forming suffixes, as well as his finding that the **-ees** and **-s** morphemes are most frequently used. Conversely, we did not find in our data significant evidence for the prominence that Kempen (1969) and Combrink (1990) give to morphemes/allomorphs such as **-der**, **-lees**, **-naar**, **-aner**, **-een**, **-ein**/**-yn** or **-ioot**; that does not mean that these do not exist – they are just not as prominent as these previous descriptions might have made us believe.

Furthermore, if we look at allomorphs due to linking elements, we identified six, viz. **-nees**, **-enaar**, **-iaan**, **-ner**, **-ter** and **-iër**. With the exception of **-nees**, all these have also been identified by Kempen (1969) and Combrink (1990). If we look closely at the instantiations of [[…] [nees]], we see that all base-form examples end on the stressed syllables [an] or [on], with the exception of *Bali* and *Mali*. A standardization body could therefore investigate whether these two examples could not be classified better under the [[…] [ër]] constructional schema, resulting in, for example, *Bali·ër*, as we also find in Dutch. If this could be the case, then it would make sense why **-nees** has not been identified by other morphologists, since it would then be a case of an allomorph due to consonant doubling, and not due to a linking element.

A similar closer look at **-ees** vs. **-nees** shows that all instantiations of the base-forms of [[…] [nees]] end on a stressed syllable, while those for [[…] [ees]] are unstressed. In the data, there is only one exception to the latter schema, viz. *Gaboen·ees* 'person from Gabon'. Since *Gaboen* ends on a stressed syllable, it would actually fit better under the [[…] [nees]] constructional schema. Support for this hypothesis comes from Donaldson (1993), where he indicates that it should be spelled *Gaboen·nees*. In the absence of usage data, and based on this categorization network, the TK could therefore reconsider the spelling of *Gaboen·ees.*

Several similar observations can be made regarding inconsistencies in the data (e.g. inconsistencies regarding base-forms ending on [stan]). In this sense, categorization networks like these could be a helpful descriptive tool for a standardization body in finding systematicity in data and rules.

6.3 Supplementary Data: Adjectival Derivations

In order to validate the generic process, the full process (as described in 4.1 and 4.2) is repeated using the supplementary data set of adjectival forms described in 5.1. Results are positive: a similarly efficient learning curve is obtained (see Figure 2) and the categorization network, although quite different, is similarly interpretable (Figure 4).

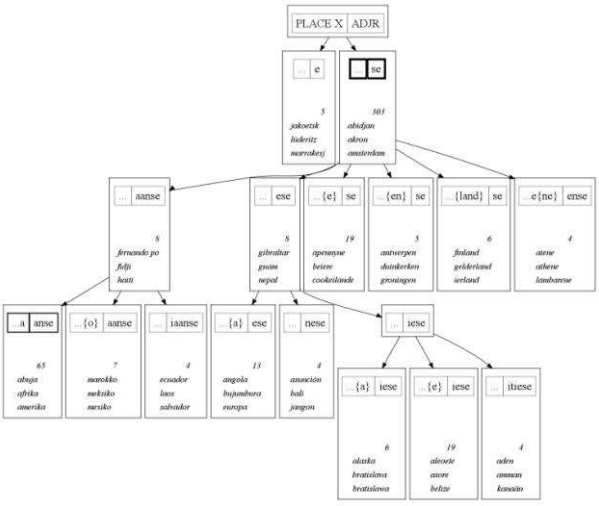

Figure 4: Categorization network for the *adjective-4* data set

7 Conclusion and Future Work

In this paper, we presented a methodology to automatically discover constructional schemas from highly irregular data, and to represent these in a way that is both interpretable by computers (predictive rule sets) and humans (categorization networks). The graphical representation is by and large compatible with one of the major Construction Grammar theories, viz. CG: we show prototypical examples (based on frequency), and also indicate relationships of elaboration. In future work, these representations could be further refined, to also indicate relationships of extensions and correspondences. We have illustrated how these representations could provide insight in our knowledge of the morphology of Afrikaans, as well as providing practical language solutions for language standardization (such as the predictor and the tables with alternative suggestions).

Other future work will continue in two directions: (1) refining the current tool for predicting derivational forms by taking additional features

into account, incorporating data that was left out in our current experiments (such as zero derivations), and benchmarking our results with regard to alternative approaches; and (2) applying our algorithm to describe other morphological constructions.

Acknowledgments

Van Huyssteen is jointly affiliated with North-West University. Support by NWU is hereby acknowledged.

Part of this research was made possible through a research grant by the South African National Research Foundation (FA207041600015).

We would like to extend our gratitude to Handré Groenewald and Martin Puttkammer for their help.

References

Combrink, J.G.H. 1990. *Afrikaanse morfologie* [Afrikaans morphology]. Pretoria: Academica.

Davel, M. & Barnard, E. 2004. A default-and-refinement approach to pronunciation prediction. *Proceedings of the 15th Annual Symposium of the Pattern Recognition Association of South Africa*. Grabouw, November 2004. pp 119-123.

Davel, M. & Barnard, E. 2008. Pronunciation Prediction with Default & Refine. *Computer Speech and Language*. 22: 374-393.

Donaldson, B.C. 1993. *A Grammar of Afrikaans*. Berlin: Mouton de Gruyter.

Goldsmith, J. 2001. Unsupervised Learning of the Morphology of a Natural Language*. Computational Linguistics* 27, pp. 153-198.

Kempen, W. 1969. *Samestelling, afleiding en woordsoortelike meerfunksionaliteit in Afrikaans* [Compounding, derivation and change of part-of-speech category in Afrikaans]. Kaapstad: Nasou.

Langacker, R.W. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford: Oxford University Press.

Shalonova, K. & Flach, P. 2007. Morphology learning using tree of aligned suffix rules. *ICML Workshop: Challenges and Applications of Grammar Induction*.

Taalkommissie. (comp.). 2009. *Afrikaanse Woordelys en Spelreëls* [Afrikaans Wordlist and Spelling Rules]. Tenth edition. Kaapstad: Pharos Dictionaries.

Tuggy, D. 2005. Cognitive Approach to Word-Formation. In: Štekauer, P. & Lieber, R. (eds.). *Handbook of Word-Formation*. Dordrecht: Springer. pp. 233-265.

Van Huyssteen, GB. 2010. (Re)defining Component Structures in Morphological Constructions: A Cognitive Grammar Perspective. In: Michel, S & Onysko, A (eds.). *Cognitive Approaches to Word-Formation*. Berlin: Mouton de Gruyter. pp. 97-126.

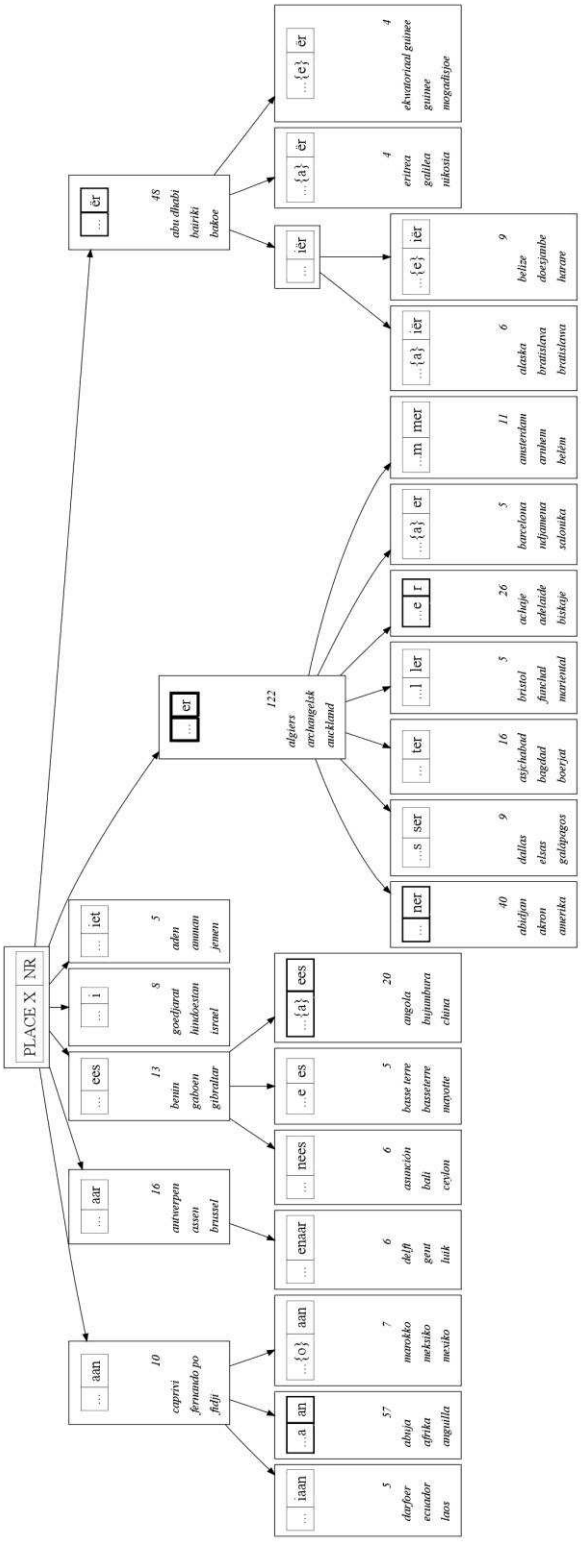

Figure 5: Categorization network for the person-4 data set