MORPA: A morpheme lexicon based morphological parser

Josée S. Heemskerk—Vincent J. van Heuven

Abstract

MORPA is a MORphological PArser developed at Leiden University for use in the text to-speech conversion system for Dutch, SPRAAKMAKER MORPA operates in three successive stages First, it generates all possible segmentations of an input word into strings of stems and affixes Secondly, it tests each segmentation for morpho syntactic well-formedness while determining word class Finally, all remaining analyses are ordered, with the most likely analysis in topmost position

In this paper we shall outline the architecture of MORPA, which comprises a dictionary of 17,087 entries, a Categorial Grammar based parser, a module for level-ordered attachment of stems and affixes and a module for likelihood determination The major problem that our system faces is ambiguity, i e , the generation of alternative segmentations and word class assignments for one input word, many of which are ungrammatical or implausible We shall discuss three kinds of strategy that have been combined in MORPA to deal with ambiguity Firstly, the system filters out ungrammatical segmentations by means of linguistic knowledge Secondly, the system orders the remaining analyses by means of frequency information And finally, morphological information that is irrelevant for pronunciation determination is eliminated In conclusion, we shall present illustrative performance data obtained from an evaluation run, involving a 3,077 word test corpus

1. Introduction

The MORphological PArser that will be described, MORPA, has been developed as a component in a text-to-speech conversion system for Dutch. Since in Dutch there is no one-to-one correspondence between orthography and pronunciation, such a system has to contain an intelligent method for converting orthography into a phonetic transcription.

As far as the pronunciation of words is concerned, we cannot just list all Dutch words with their pronunciation in a dictionary, since the speakers of the language have the possibility to create new words and the vocabulary, as such, is indefinitely large. Nominal compounding, e.g., is a highly productive morphological process in Dutch. Moreover, new compounds are spelled without spaces or hyphens between their parts, as in (1):

(1)	a.	woord	'word'
	b.	woordgrens	'word boundary'
	c.	woordgrenssymbool	'word boundary symbol'

The alternative way to obtain the phonetic transcription is the phonological-rule approach. Dutch phonological rules are dependent in several ways on morphemic segmentation and word class assignment. For example, as shown in (2a), the graphemes *d* and *b* are pronounced voiceless when they occur stem-finally, but voiced when they occur stem-initially. As shown in (2b) stress in compounds differs from stress in monomorphemic words. In (2c) it is shown that stress in (predicatively used) adjectival compounds differs from stress in nominal compounds:

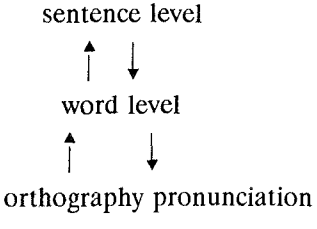
(2)	a.	hoofdagent	hoof[t] + agent	'police sergeant'
		loofdak	loof + [d]ak	'roof of foliage'
		krabijzer	kra[p] + ijzer	'scratch iron'
		slaboon	sla + [b]oon	'butter bean'
	b.	avonduur	'avond + uur	'evening hour'
		avontuur	avont'uur	'adventure'
	c.	onecht	on + 'echt, Adj	'unreal'
		onrecht	'on + recht, N	'injustice'

It will be clear from these examples that if the text-to-speech system is to produce high quality speech it must be provided with a module for morphological analysis that recovers morphemic segmentation and word class.

MORPA performs only part of the linguistic processing that is necessary to convert orthography into a phonetic representation. In the linguistic processing of the input text two aspects are of importance:

I. The morphological structure of words and the syntactic structure of sentences have to be obtained;
II. The morphological structure of words has to be interpreted in order to be able to assign a phoneme representation, syllabication and stress. The syntactic structure has to be interpreted in order to determine the place of accents and pauses.

Schematically,



Within SPRAAKMAKER the distribution of tasks is as follows:

Orthography →	Word Level:	MORPA	
Word level ↔	Sentence Level:	PROS2	(cf. Dirksen—Quené, this volume)
Word Level →	Pronunciation:	MORPHON	(cf. Nunn—van Heuven, this volume)

MORPA provides word structure and word class. The word class is used by PROS2 in order to build syntactic structure and both word structure and class are used by MORPHON to determine syllabification, stress and phoneme transcription.

2. Architecture of MORPA

Within MORPA two kinds of modules are distinguished: the morphological modules for morphological analysis and the application modules that have been created with a view to implementation in the text-to-speech-system. Section 2.1 will describe the morphological modules; section 2.2 will describe the application modules.

2.1. Morphological modules

Figure 1 shows the structure in which the morphological modules operate. The module MORPA is the coordinating module that delegates the main tasks. Within module MORPA the morphological analysis of an input word is carried out in three successive stages. First, the word is segmented into a string of stems and affixes. This task is delegated to the SEGMENTATION module. In the second stage each segmentation is checked for its grammaticality by the PARSING module, and, if it is grammatically well-formed the word class and the likelihood of the analysis are determined. Finally, all remaining analyses are ordered on the basis of their likelihood. This task is not delegated but performed by module MORPA. Below we shall discuss each stage and go into the details of the module structure.

2.1.1. Segmentation module

The SEGMENTATION module divides the input word into all possible substrings. Each substring is submitted to the module MORPHEME which checks whether it is a Dutch morpheme or not. In order to find out whether a substring is a morpheme or not, the module MORPHEME uses (1) a LEXICON, which contains a large number of morphemes, (2) the module

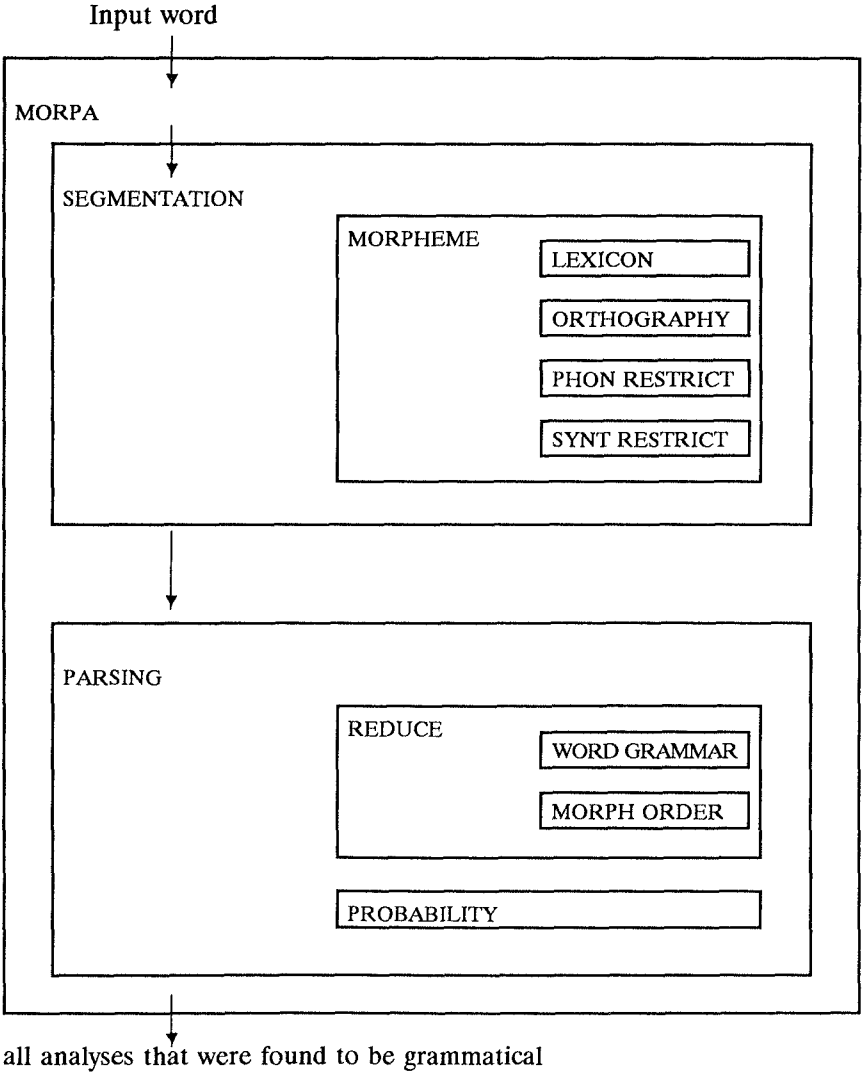


Figure 1. Global functional architecture of MORPA

ORTHOGRAPHY, which has knowledge of the orthographic rules that have to be taken into account, (3) the module PHON RESTRICT, which tests for some phonetic restrictions and (4) the module SYNT RESTRICT which tests for some preliminary syntactic restrictions. Each of the modules that are used by MORPHEME, are discussed in this section.

Our method for morphological analysis comprises a LEXICON. In this

approach the word or word parts are recognized through dividing the word into substrings that correspond to entries in the lexicon.

Each entry in the lexicon contains the following fields of information:

I. *Orthographic form.* This form is the key to recognition. Since many morphological processes are highly productive, we cannot just list all Dutch words with their relevant information in a lexicon. We can, however, list all word formations that belong to closed classes, along with all simplex words and productive affixes, which are potential ingredients of newly formed words.[1] In that case the morphological parser only has to analyze words formed according to productive rules. The notion "potential ingredients of newly formed words" is a pragmatic operationalization of the theoretical notion "morpheme", which is traditionally defined as "the smallest meaningful part in word formation".

II. *Morpho-syntactic properties.* In this field the following syntactic categories are distinguished: N(oun), V(erb), Adj(ective), Adv(erb), Prep(osition), Det(erminer), Pro(noun), Q(uantifier), Conj(unction), Int(erjection) and P(roper)N(name). For the benefit of inflection (MORPA) and syntactic analysis (PROS2) some of these categories are subdivided on the basis of inflectional and selectional features. See (3) for some examples:

(3) a. [orth = *huis,* cat = n (n, *en,* c)] 'house'
b. [orth = *zeg,* cat = v (stem (h), [i (*te*), f (*dat*)]] 'say'
c. [orth = *ver,* cat = n (_, _, _) \v (stem (z), [ ])] PREFIX

In (3) the orthographic form *huis* is morpho-syntactically specified as noun(neuter, plural: *en*, count_noun); *zeg* is specified as verb(stem(auxiliary: *heb* 'to have'), [infinitival_complement_with(*te* 'to'), finite_ complement_with(*dat* 'that')]. As we shall discuss in section 2.2. the complex categorial representation of the affixes, such as *ver*, comprises the word grammar implemented. From the set of word formation rules the improductive rules were excluded.

III. *Morphological classification.* This field classifies the morpheme for purpose of stress assignment. The labels comprise notions such as prefix, stem or suffix; native or non-native; stress attracting, stress neutral or stress bearing, etc. (See section 2.2. and Nunn—van Heuven, this volume, for motivation and application).

IV. *Phonological form.* Each lexical entry has been provided with an underlying pronunciation. After analysis by MORPA, MORPHON uses underlying pronunciation, morphological classification and word category to determine the surface pronunciation of complex words (cf. Nunn—van Heuven, this volume).

V. *Frequency information.* In order to be able to determine the likelihood of an analysis on the basis of the likelihood of the constituents each lexical entry is provided with frequency information, viz. the logarithm of the probability that the morpheme is member of its syntactic class. (See section 2.1.2. for a description of likelihood determination).

VI. *Morpho-phonological peculiarities.* Three optional fields contain codes that indicate (a) phonetic restrictions on the recognition of specific suffixes, (b) for some suffixes to trigger orthographic rules, (c) for some morphemes to be marked as exceptions to an orthographic rule. Below we shall discuss the orthographic rules and phonetic restrictions.

At present, the lexicon contains 17,087 entries: 12,264 simplex words, 468 affixes and 4,355 improductively formed complex words.

The ORTHOGRAPHY module takes into account that Dutch word stems, when inflected or used as the basis of a derivation, may undergo spelling changes. See e.g. the words in (4) that undergo (a) vowel degemination, (b) consonant gemination and (c) voicing of the stem-final *s* when pluralized:

(4)	a.	baan	banen	'jobs'
	b.	man	mannen	'men'
	c.	laars	laarzen	'boots'

When segmenting, MORPA has to undo the spelling changes in order to recover the stem from the plural and be able to access it in the lexicon. In the morpho-phonological field of the lexicon a number of suffixes are marked as potential triggers of spelling changes. Since these triggers always follow the stem that undergoes the spelling change, we decided to make our segmentation procedure operate from right to left: only if a potential trigger is found, does MORPA attempt to apply the spelling rules (in reversed direction) to the remainder of the input word. The spelling rules are embedded in our Prolog program; see for a more principled account of spelling changes in morphological parsing Koskenniemi (1983).

The module PHON RESTRICT tests whether there are phonological or phonetic restrictions on the recognition of the morpheme in a specific context, i.e., following a phonologically specified left neighbor. For instance, the agent suffix *-aar* is an allomorph of the suffix *-er* which only occurs when following a so called "schwa-stem", i.e. a stem ending in /-əl/, /-ən/ or /-ər/. Therefore, *-aar* is accepted in (5a) but rejected in (5b). The stem *werk* 'work' takes the suffix *-er* (5c):

(5)	a.	wandelaar	wand[əl] + aar	'walker'
	b.	werkaar	*werk + aar	
	c.	werker	werk + er	'worker'

This test is initiated by a special code in the lexical representation of the morpheme.

The module SYNT RESTRICT tests on some preliminary syntactic properties in order to prevent the parsing module from doing work of which we know beforehand that it will be in vain. The two main tests are:

I. Test on open class. Only morphemes that belong to open classes (N, V, Adj, Adv) and affix-categories are allowed as word parts. Morphemes that belong to closed classes are rejected beforehand.
II. Test on reduplication. In Dutch reduplication, i.e., the occurrence of a specific morpheme twice in succession, is not allowed. A morpheme that is recognized a second consecutive time is excluded.

To illustrate the effect of the segmentation procedure, its output for the noun *beneveling* 'intoxication' is shown in (6):[2]

(6) a. be + neef + eling
b. be + neef + e + ling
c. be + nevel + ing
d. been + e + veel + ing
e. be + n + e + veel + ing
f. be + neef + eel + ing

The segmentation procedure analyses the input word into all possible strings of stems and affixes without any further grammatical knowledge. Thus, along with the plausible segmentation *be + nevel + ing*, (6c), several alternative segmentations are generated which violate grammatical and/or semantic restrictions. So, in order to filter out the ungrammatical segmentations and determine the word class of the grammatical ones, each segmentation must be checked for its grammaticality. Therefore each segmentation is submitted to the PARSING module that contains a categorial parser (which may be viewed as a word grammar)[3] and rejects every segmentation that is not in accordance with the rules of Dutch morphology. This categorial parser is discussed in the next subsection.

2.1.2. The parsing module

In the PARSING module all the segmentations that are generated by the SEGMENTATION module are tested for their morpho-syntactic well-formedness. Of each grammatical segmentation word class and likelihood are determined.

A segmentation is found to be well-formed if the string of constituents can be reduced to one constituent. Therefore the PARSING module recursively submits two constituents to the module REDUCE, which will try to reduce them to one. Two constituents can be reduced if the combination is in accordance with the rules for Dutch word formation. These rules are applied by the modules WORD GRAMMAR and MORPH ORDER. If a reduction has succeeded, the syntactic class of the new constituent is derived and the PARSING module will activate the module PROBABILITY in order to determine the likelihood of the constituent. In this section we will discuss the modules WORD GRAMMAR, MORPH ORDER and PROBABILITY.

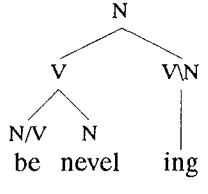
Our word grammar is designed according to the principles of Categorial Grammar (cf. Moortgat 1987). As a consequence all relevant morpho-syntactic information is represented in the lexicon while the syntax, which is implemented in the module WORD GRAMMAR, merely consists of three simple reduction laws.

In our lexicon, all affixes are assigned a complex category. That is, prefixes have been assigned a category of type X/Y, which means that they take a stem of category X on their right-hand side to yield a word of category Y. For instance, the prefix *be-* with category N/V demands a nominal stem to the right to form a verb. Likewise, suffixes of category X\Y look for a stem of category X on their left-hand side to combine to a word of category Y. Thus, the suffix *-ing*, V\N, demands a verbal stem to the left to form a noun. Two adjacent stems XY may, according to the Right Hand Head Rule, be combined into a word of category Y.[4]

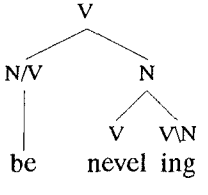
The reduction laws are iteratively applied to adjacent categories in a strictly bottom-up fashion. An analysis fails as soon as a string of categories cannot be reduced to one single category.

The examples in (7) illustrate how iterative categorial reduction results in a successful parse. The structures show the derivation and determination of the output category of (6c).

(7) a.



b.



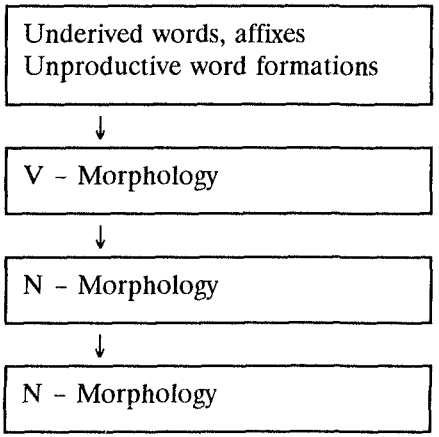
The ambiguity in (7) is due to the fact that the morpheme *nevel* has more than one lexical category and as a consequence can be reduced in more than one way, resulting in an incorrect word class assignment in (7b).

Obviously, the word grammar is not restrictive enough. In order to be capable of excluding analyses like (7b), the parser is supplemented with a component for attachment ordering. This module imposes an ordering on the attachment of affixes and stems which restricts the type of stem that an affix or stem may attach to.

The module for attachment ordering, MORPH ORDER, is inspired by Lexical Morphology and by one lexical theory for Dutch in particular (cf. van Beurden 1987). The model implemented is an extension of van Beurden's model in a way which is consistent with its basic assumptions.[5]

According to this theory the Dutch vocabulary can be divided into four levels, as in (8), which may be viewed as possible successive stages in word formation. The first level, or lexical level, comprises our lexicon of simplex words, affixes and irregular formations. This level also contains all (borrowed) Romance words. The elements of this lexical level may be successively developed on the second level on which V(erbal)-morphology takes place; the third level on which A(djectival)-morphology takes place and the fourth level on which N(ominal)-morphology takes place. Each of these levels preserves the possibility for suffixation, compounding and prefixation. On the levels for V-morphology and A-morphology each of these processes may take place only once. We assume that only the processes on the N-morphology level are recursive, i.e., may take place more than once. (cf. Heemskerk 1989 for more detail).

(8)



With respect to the word *onverdraagzaamheid* 'intolerance' in (9) the model correctly predicts that verbal prefixation precedes adjectival suffixation and prefixation, which, in turn, precedes nominal suffixation:

(9) [[on_{pref} [[ver_{pref} $draag_{stem\ V}$] $zaam_{suf\ Adj}$] $_{Adj}$] $heid_{suf\ N}$]

This level module rules out the analysis in (7b): the nominal suffix *-ing* may not be attached before the verbal prefix *be-*. Therefore the word cannot be analyzed as a verb.

At this point, we return to the example of (6). Of the six alternative segmentations, four (10a-d) are accepted by the categorial component. The assigned word classes are given in (10i). The level module rejects (10b) and (10d) and the verbal derivation of (10ci), as shown in (10ii):

(10)	a.	be + neef + eling	i) N	ii) N
	b.	be + neef + e + ling	N	—
	c.	be + nevel + ing	N V	N
	d.	be + neef + eel + ing	N	—

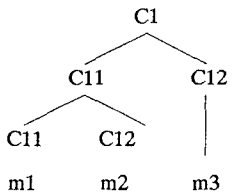
This leaves us with two analyses; the first of which (10aii) is implausible; the second of which (10cii) is correct.

Clearly, the ultimate handling of the remaining ambiguity demands recourse to semantics and world knowledge. For pragmatic reasons, however, we explored several different strategies for ordering analyses in terms of plausibility, aiming at a system that generates the best analysis first. If the correct analysis is not generated in first position, as in (10), the word may receive a wrong pronunciation.

The ordering of the analyses in (6) and (10) is imposed by two preliminary criteria: firstly, in our right-to-left segmentation procedure priority is given to the longest morpheme in the dictionary that matches the input, an idea that has been suggested by several authors for various languages (cf. Allen—Hunnicutt—Klatt 1987; Daelemans 1987). Secondly, whenever an affix has alternative category assignments, the one with the highest lexical frequency is tried first. As we can conclude from (10) these criteria do not always succeed in selecting the best analysis out of the set. Therefore, we improved the ordering strategies by employing both lexical and text frequencies of morphemes, categories and category transitions. The heuristic strategy which is implemented in the module PROBABILITY, orders the competing analyses along a scale of plausibility and replaces the criteria mentioned above. We shall describe the ordering strategy below.

Suppose that MORPA proposes the analysis in (11) for a word that is the concatenation of the morphemes m1, m2 and m3:

(11)



In order to determine the probability of (11) we need the following information:

I. The probability that the word belongs to the syntactic category C1 that MORPA proposes. In formula:

$$p(WCn) = \frac{\#(WCn)}{\#(W)}$$

Where:

W	= the set of words;
WCn	= the subset of words from W that are of category Cn;
p(WCn)	= the probability that a random word in W is of category Cn;
#(WCn)	= the number of words in WCn;
#(W)	= the number of words in W.

II. The probability that a word has the hierarchical structure that MORPA proposes. We consider a hierarchical structure to be a series of transitions from mother to daughter categories. In (11) the mother category C1 branches into the left-hand daughter category C1 and the right-hand daughter category C2. These daughters are called C11 and C12 to indicate that they originate from mother category C1. Daughter category C11, in turn, branches into the daughter categories C1 and C2 and these categories are called C111 and C112 to indicate that they originate from C11.

In order to determine the probability of this structure we do not take into account the position of every category in the structure. Rather, we consider the transitions C1 → C11, C12 and C11 → C111, C112 to be identical, i.e., embeddedness is not taken into account.[6] The probability of the structure is the product of the probabilities of the separate transitions. In formula the probability of a transition Cn → Cn1, Cn2:

$$p(TCnCn1Cn2) = \frac{\#(TCnCn1Cn2)}{\#(TCn)}$$

Where:

T	= the set of mother-daughter category transitions;
TCn	= the subset of transitions from T with mother category Cn;
TCnCn1Cn2	= the subset of transitions from TCn, with left-hand daughter category Cn1 and right-hand daughter category Cn2;
p(TCnCn1Cn2)	= the probability that a random transition of TCn is Cn → Cn1, Cn2;
#(TCnCn1Cn2)	= the number of transitions that are in TCnCn1Cn2;
#(TCn)	= the number of transitions that are in TCn;

III. The probability that a word contains the morphemes that MORPA proposes. In (11) the analysis comprises the morphemes m1 of category C(11)1, m2 of category C(11)2 and m3 of category C(1)2.

The probability that an analysis contains m1, m2 and m3 is the product of all the individual morpheme probabilities. In order to determine the probability of a morpheme its position (leftmost, rightmost, middle) in the analysis is not taken into consideration.[5] In formula:

$$p(m|MCn) = \frac{\#(m|MCn)}{\#(MCn)}$$

Where:

M	= the set of morphemes;
MCn	= the subset of morphemes from M that are of category Cn;
p(m|MCn)	= the probability that a random morpheme from MCn is m;
#(m|MCn)	= the number of morphemes m in MCn;
#(MCn)	= the number of morphemes in MCn.

Then, the probability of an analysis is the product of the probabilities of the syntactic category, the structure and the morphemes. For instance, the probability of (11) is:

p([C1 [C11 [C111, m1],[C112, m2]], [C12, m3]]) =

p(C1) *	(syntactic category)
p(C1,[C11,C12]) * p(C11,[C111,C112]) *	(transitions)
p(m1|C111) * p(m2|C112) * p(m3|C12)	(morphemes)

The set of words W on which our determination is based is the CELEX-database that contains approx. 123,000 Dutch stems provided with syntactic information, a morphological decomposition and token frequency information (cf. van der Wouden 1988). The frequency information is based on a 44 million words corpus. The morphological decomposition and frequency information were used to extract mother-daughter category frequencies and morpheme frequencies. The syntactic category and frequency information were used to extract word category frequencies.

From our database we simultaneously collected lexical (type) frequencies and usage (token) frequencies. Lexical frequencies are extracted from a dictionary and usage frequencies are extracted from a text corpus. Therefore, the module PROBABILITY exists in two versions: the version that is based on lexical frequencies is used for tests on dictionary samples in which every word is analyzed by MORPA once; the version that is based on usage frequencies is used for tests on text samples in which every word is analyzed by MORPA as many times as it occurs in the text.

In conclusion let us return to the example *beneveling* of (10). After determination of the likelihood of the remaining analyses (10a) and (10c) by the module PROBABILITY the correct analysis (10cii) *be* + *nevel* + *ing* is correctly ordered in topmost position:

(12) a. *be* + *nevel* + *ing* , N
b. *be* + *neef* + *eling*, N

In section 3 we shall give performance data which are based on a test sample containing 3,077 test words.

2.2. The application modules

Several modules have been created for the purpose of implementing MORPA in a text-to-speech system. These modules take care of the communication of the morphological parser with external modules that precede or follow MORPA within the text-to-speech system. Here we shall describe the interfaces to TEXT-SCAN, the preceding module and MORPHON, one of the following modules.

TEXT-SCAN is a module that preprocesses text in order to make text containing symbols, abbreviations, acronyms, proper names, etc. suitable for linguistic analysis. For this purpose the input text is segmented into sentences, and words or phrases are labeled. Each label refers to the status of the segment, e.g., *acronym, proper name, punctuation, abbreviation, number,* etc. (cf. van Holsteijn, this volume). MORPA is unable to analyze non-lexical items such as numbers, acronyms and proper names, because

they belong to very large classes of which complete storage in the morpheme lexicon is impossible. Therefore, MORPA will only be presented text segments that are labeled "lexical item". Then, words containing acronyms (13a), numbers (13b) or proper names (13c) are problematical because they have to be analyzed by MORPA, but contain elements that MORPA is not able to recognize.

(13) a. HTS-er, CAO-overleg
b. 18-jarige, begroting-1991
c. Madonna-rage, Gorbatchov's

In order to be able to analyze them properly, TEXT-SCAN labels each part of these words and assigns the label "lexical item" to the whole word:

(14) CAO-overleg ⟶
lexical(acronym(*CAO*), punctuation(-), lexical(*overleg*))

In this way, MORPA is able to interpret the various TEXT-SCAN labels and act on them. In order to deal with the non-lexical items they are added to a temporary lexicon to which the morphological modules of MORPA have access. In this lexicon the non-lexical items have a default specification: acronyms are specified as noun, digits as numeral and proper names as proper names. The only thing the temporary lexicon cannot provide is a phonological form. A module following MORPA provides for it (cf. te Lindert—van Leeuwen, this volume).

MORPHON is the module that contains phonological rules that derive a pronunciation representation (cf. Nunn—van Heuven, this volume). As we have seen in section 1, phonological rules are dependent in several ways on morphemic segmentation and word class assignment. Therefore MORPA must precede MORPHON. The output of MORPA, however, contains some information that is irrelevant to pronunciation determination. These overspecifications are eliminated since they give rise to unnecessary ambiguity, which reduces the chance for the correct analysis to end up in topmost position and, as a consequence, for the word to receive its correct pronunciation.

To some extent the categorial labeling of morphemes is redundant. It appears that for correct assignment of pronunciation, the morphological classification, i.e., prefix-, stem- or suffixhood of the morphemes, is more crucial than categorial classification. For instance, in (15) the verbal stem *verwerk* 'to process' has been assigned two analyses:

(15) a. $[_V\ [_{N/V}\ ver],\ [_N\ werk]]$
b. $[_V\ [_{V/V}\ ver],\ [_V\ werk]]$

Although the analyses in (15) differ in the labeling of the morphemes, both receive the same, correct, pronunciation /vər-ˈwɛrk/. This does not hold for the analyses in (16):

(16) a. [$_{V}$ [$_{Adj}$ *ver*], [$_{V}$ *spring*]]
b. [$_{V}$ [$_{V/V}$ *ver*], [$_{V}$ *spring*]]

According to (16a) the word, meaning 'do the long jump', is pronounced /ˈvɛr-sprɪŋ/, whereas (16b), meaning 'change suddenly', is pronounced /vərˈsprɪŋ/. This difference in pronunciation arises from the fact that in (16a) *ver* 'far' is a stem, and as a consequence the word a compound, whereas in (16b) *ver* is a prefix and the word a derivation.

There is also a measure of overspecification in the hierarchical morphological structure that reflects the derivation. It may play a role in compound stress assignment, but its effect is marginal. Moreover, hierarchical structure can only be used for stress assignment on the condition that MORPA presents the correct structure. In practice, however, MORPA offers more than one structure and for choosing the right structure semantics has to be taken into account. Consider e.g. the analyses of the word *paardenfokkerij*. The analysis in (17a) is pronounced with primary stress on the last syllable and corresponds to the meaning of 'horse-breeding'; the analysis under (17b) is pronounced with primary stress on the first syllable and could be said to correspond to the meaning of 'stud-farm'. MORPA generates both analyses, but will not be able to choose between them for other reasons than likelihood. Also on the basis of likelihood, MORPHON will assume that the left-branching analysis (17a) is correct:

(17) a. [$_{N}$ [[$_{N}$ *paarden*], [$_{V}$ *fok*]], [$_{V\backslash N}$ *erij*]]
b. [$_{N}$ [$_{N}$ *paarden*], [[$_{V}$ *fok*], [$_{V\backslash N}$ *erij*]]]

Morphemic segmentation and overall category are relevant information. Consider, e.g., the word *balletje* in (18). The analysis in (18a) corresponds to the meaning 'ball + DIM(imutive)' and is pronounced /ˈbɑl-lə-tjə/; the analysis in (18b) means 'ballet + DIM' and is pronounced /bɑl-ˈlɛt-jə/. Here, the difference in pronunciation is due to different segmentations:

(18) a. [$_{N}$ [$_{N}$ *bal*], [$_{N\backslash N}$ *etje*]]
b. [$_{N}$ [$_{N}$ *ballet*], [$_{N\backslash N}$ *je*]]

The word *apetrots* in (19) is also ambiguous in pronunciation: the analysis under (19a) corresponds to the meaning 'pride of a monkey' and is pronounced /ˈa:-pə-trɔts/; the analysis under (19b) means 'proud as a monkey'

and is pronounced /a:-pə-'trɔts/. Here, the difference in stress is due to different overall category:

(19) a. $[_{N}\ [[_{N}\ aap],\ [_{N\backslash N\text{link}}\ e]],\ [_{N}\ trots]]$
b. $[_{Adj}\ [[_{N}\ aap],\ [_{N\backslash N\text{link}}\ e]],\ [_{Adj}\ trots]]$

Consequently, the output of the morphological modules undergoes two (ordered) operations in order to yield an output that contains only information that is relevant to pronunciation:

I. *label-conversion*, i.e., each lexical category label is replaced by a label that is in the lexical representation of the morpheme and that reflects the morphological classification of the morpheme. For purpose of the determination of the pronunciation this notion "morphological class" is enriched with distinctions such as native/non-native, stress-neutral/-stress-attracting/stress-bearing etc.[7]

II. *linearization*, i.e., all alternatives which have the same segmentation (i.e., string of labeled morphemes) as well as the same overall category are collapsed into one linear representation. Thus, (15a) and (15b) are collapsed into (20a) and (17a) and (17b) are collapsed into (20c) as they contain unnecessary, redundant information. The analyses of (16), (17) and (18), which do not contain irrelevant information, are still distinguished in (20b), (20d), and (20e):

(20) a. $[_{V}\ [_{pref}\ ver],\ [_{stem}\ werk]]$
b. $[_{V}\ [_{stem}\ ver],\ [_{stem}\ spring]]$
$[_{V}\ [_{pref}\ ver],\ [_{stem}\ spring]]$
c. $[_{N}\ [_{stem}\ paarden],\ [_{stem}\ fok],\ [_{suf}\ erij]]$
d. $[_{N}\ [_{stem}\ bal],\ [_{suf}\ etje]]$
$[_{N}\ [_{stem}\ ballet],\ [_{suf}\ je]]$
e. $[_{N}\ [_{stem}\ aap],\ [_{link}\ e],\ [_{stem}\ trots]]$
$[_{Adj}\ [_{stem}\ aap],\ [_{link}\ e],\ [_{stem}\ trots]]$

Through converting output that contains hierarchical structures and categorial labels to linear structures and morpheme classification labels much unnecessary ambiguity is eliminated.

3. The performance of MORPA

In order to evaluate the performance of our system in the state described above and with output as described in section 2.2., a lexical test was run on

a dictionary sample of 3,077 words, taken from texts of the so called "PB corpus" (cf. Bringmann 1990). Figure 2 shows the test results.

As is shown in Figure 2, MORPA assigns no analysis at all to 3 percent of the test words. Ninety-six percent of the test words were assigned a correct analysis, but for 52 percent of the test words the correct analysis was among alternatives; the average number of analyses for the whole test being 2.0. For 48 percent of the test words the correct analysis was generated as the most likely member of a set of alternatives. To 44 percent of the sample MORPA assigned a single, correct analysis. In sum, 92 percent of the test words received a correct analysis that is generated in first place.

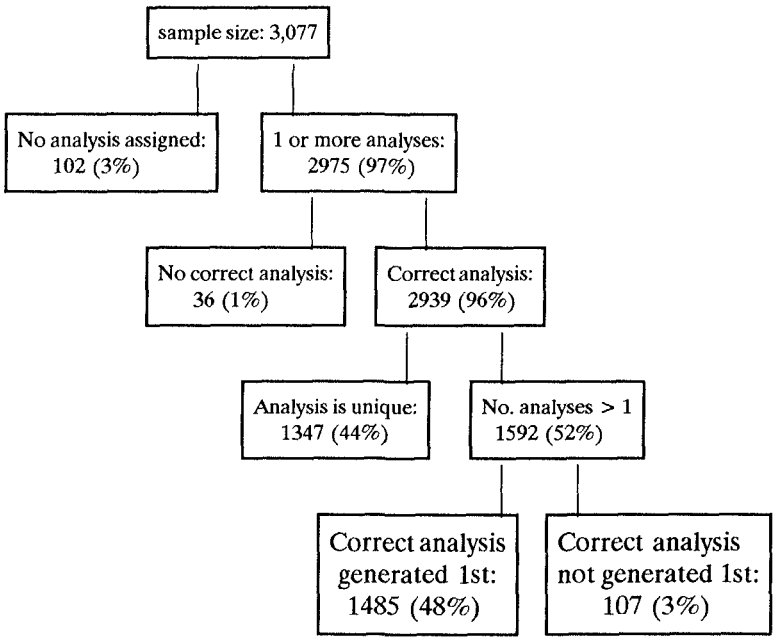


Figure 2. Summary of the test results; all percentages are expressed relative to N = 3,077.[8]

At the time that MORPA was only provided with a segmentation module and a parsing module that only consisted of the module WORD GRAMMAR the correct analysis was generated as the first or only alternative for 79 percent of the test words (cf. Baart—Heemskerk 1988). After addition of the module MORPH ORDER this percentage had risen to 85 percent (cf. Heemskerk 1989). In these stages MORPA contained the preliminary criteria that imposed an order on the set of alternative analyses. The

eventual rise to 92 percent is due to the substitution of the module PROBABILITY for the preliminary ordering criteria.

As to the relevance for word level pronunciation, it was established that 82 percent of the errors made by MORPA led to an incorrect pronunciation representation (cf. Nunn—van Heuven, this volume).

4. Conclusions

For high quality text-to-speech conversion in Dutch a module for morphological analysis is needed. The morpheme lexicon approach is a principled one, but a major problem that it faces is ambiguity. In this chapter we have shown that within MORPA the ambiguity problem is largely reduced or sidestepped: MORPA has a success rate of 92 percent. In the first place ambiguity was reduced by implementing a word grammar and limiting the analysis to productive word formations. Next, ambiguity was reduced by implementing a module that defines the order in which morphemes may be attached to stems. Ambiguity was further reduced through eliminating information that is irrelevant for pronunciation. Finally, an heuristic strategy was implemented that orders the set of remaining alternatives along a probability scale.

Acknowledgements

This project was carried out by the first author, under the supervision of Joan Baart and the second author. The morpheme lexicon was compiled with the assistance of Marjan Grootveld, Brit van Ooijen, Jenny Doetjes, and Cecile Davis. Programming assistance came from Maarten Hijzelendoorn, Pieter Toussaint, and Jos Pacilly.

Louis ten Bosch' help on the construction of the probability module is gratefully acknowledged.

Finally, we thank all participants of the theme group "Linguistic Analysis" for their patient help, recommendations, support and fine cooperation during the project.

Notes

1. Because we assume word formation to be word based, our lexicon does not contain roots.
2. This example only shows a small number of segmentations. With increasing lexicon size and decreasing average length of the lexical elements, the number of alternative segmentations

for an input word grows A combinatorial explosion can be the result, causing hundreds of segmentations to be generated

3 Note that here "word grammar" is used in the sense of "a set of word formation rules", and is not to be confused with Richard Hudson's "word grammar"

4 The rules for prefixation and suffixation are functional application laws The law for compounding is not a categorial law, it is the implementation of the Right Hand Head Rule (cf Trommelen—Zonneveld 1986)

5 Because we are mainly interested in the morphological aspects of the proposed level model, we shall leave the phonological claims for what they are

6 The position of a transition or morpheme is not taken into account for two reasons (a) the determination of dependent probabilities would result in very large tables, (b) every structure that is not in the corpus W would receive a probability $p = 0$ with the result that newly created words are always ordered in last position

7 For the sake of clarity, these labels are not mentioned in the examples

8 Because the percentages are rounded off they do not add up to 100 percent