Automatic Hyphenation of Dutch Words based on Linguistic Rules

Anneke Nunn
Van Dale Lexicografie, P.O. Box 19232, 3501 DE Utrecht, The Netherlands

Abstract

This paper describes a method (reverse engineering) to improve the quality of hyphens in a dictionary database. Hyphens are recomputed with spelling-based linguistic rules. Since the input of the hyphenation program is supplied with high-quality lexicographic information including morphological make-up, good results can be obtained with a simple algorithm without compound analysis. These results could not have been achieved with earlier hyphenation programs based on word lists. The current method also has advantages over earlier hyphenation programs based on phonological syllable structure.

Traditionally, the compiling of dictionaries has been the work of lexicographers who laboriously and conscientiously record words with their meaning, usage and formal features such as spelling with hyphens, inflection and pronunciation. Particularly with respect to formal features, however, computers have two advantages over human editors: when provided with a correct algorithm, they can calculate these features rapidly and efficiently for large quantities of words, and they can do so without inconsistencies or errors. By using the computer, dictionary makers can leave the chore of describing regular words to the computer and concentrate on rules and exceptions.

In this paper, I will concentrate on the automatic generation of one formal feature of Dutch words: the spelling with hyphens, i.e. marks that indicate where words can be divided at the end of a line. It will be argued that all existing hyphenation algorithms have serious disadvantages, so that a new program had to be written that is suitable to compute the hyphenation pattern of dictionary entries. A simple hyphenation program without morphological decomposition suffices, since its input consists of words with high-quality lexicographic information which includes morphological make-up. The paper is organized as follows: Section 1 explains why a new hyphenation program was developed. Section 2 summarizes some earlier proposals. Section 3 describes the Dutch hyphenation rules and their background. Section 4 describes the implementing and testing of the hyphenation program. Section 5 discusses the differences of hyphenation positions added by editors and those computed by the program. Finally, the paper ends with some conclusions and suggestions for further improvements of the algorithm. [1]

1 Motivation for the development of a new hyphenation program

Van Dale Lexicografie publishes several Dutch and bilingual dictionaries. Formal features of the entries of these dictionaries such as spelling, spelling with hyphens, inflection and pronunciation are extracted from a product-independent, central database. Until recently the bulk of the information in this database was compiled and edited by hand. This also holds for the spelling with hyphens, which is the subject of this paper. However, after some time it became apparent that this method had some serious disadvantages.

Obviously, this method is very time consuming since every word must be provided with different syllable markers (to be discussed below). An improvement was obtained by break-

[1] Following the typographical conventions of Dutch dictionaries, hyphenation positions will not be marked by hyphens but by dots, to distinguish them from hyphens that are used to join two words. The word *niet·ro·ker* ('non-smoker'), for instance, is always written with one hyphen; a second one is only inserted when the word is divided.

ing up compounds into their constituting parts and hyphenating all different parts only once, e.g. *avond* ('evening'), *sche·mer* ('dusk') rather than hyphenating all simplex words and compounds, e.g. *avond*, *sche·mer*, *avond·sche·mer* ('twilight'), *sche·mer·avond* ('dusky evening'). Derivations and inflected forms, however, must all be handled individually because of resyllabification: *sche·me·rig* ('dusky'), *avon·den* ('evenings'). This inefficiency is becoming more urgent with the rapid increasing of the size of the database which is currently taking place: for the electronic version of the *Grote Van Dale* (large Van Dale) which will appear shortly ± 250.000 entries have to be hyphenated. In addition, all possible inflected forms of each entry must be hyphenated as well.

These forms consist of the diminutive and plural forms of nouns, the inflected form, comparative and superlative of adjectives and their inflected forms, and the complete verbal paradigm consisting of twelve verb forms. To create this enormous amount of hyphenated words by hand would cost a lot of time. Furthermore, after the database has been created it has to be maintained, which is not efficient either: a small change in the hyphenation conventions would necessitate a new round of editing.

Another, more serious disadvantage is that hyphenating by hand may lead to inconsistencies and errors. The Dutch hyphenation rules are not very explicit. Editors who hyphenate words based on these rules may interpret rules differently. This means that it is probable that some word types are not treated consistently. Furthermore, editors may make mistakes, but so few that they are hard to find. By using a computer program these disadvantages are avoided. After all, the rules must be made explicit in order to make the algorithm, so hyphenation becomes both consistent and reproducible.

2 Hyphenation algorithms

A hyphenation program that can be used in the dictionary database must fulfil the following requirements: in the first place, it must generate the most recent hyphenation patterns, since a recent spelling reform in 1995 affected some aspects of hyphenation (see below). Secondly, it must generate all hyphenation positions of words. Thirdly, it must formalize hyphenation in an insightful manner so that the rules may be easily understood and adjusted if necessary. Preferably this goal should be achieved by using rules that imitate the official hyphenation rules in order to eliminate the possibility of mismatches between hyphenation rules and hyphenated words.

In the literature, two types of algorithm have been proposed for Dutch: those based on linguistic rules, e.g. Daelemans (1987, 1989), and those based on word lists, e.g. Brandt Corstius (1970) or pattern matching, e.g. Boot (1984), Tutelaers (1996), which is a Dutch version of Liang (1989). These hyphenation algorithms do not fulfil the requirements mentioned above. In the first place, they generate the hyphenation patterns from before the spelling reform of 1995. Secondly, they do not always generate all possible hyphenation positions. For instance, in Daelemans' algorithm hyphenation positions that are two letters away from the word edge are ignored, for typographical reasons. Finally, the methods used deviate quite significantly from the official hyphenation rules. I will briefly discuss the disadvantages of the two types of proposals.

Hyphenation programs based on word lists will treat all words which are in the list correctly. However, as already extensively argued by Daelemans, they will fail for many new words. The reason is that the morphological make-up of words influences their hyphenation, cf. for instance the underived word *lui·ster* ('lustre') and the word *lui·ste* ('laziest') which contains the superlative suffix *-ste*. Compound boundaries are also crucial for hyphenation, cf. the compound *min·acht* ('disdain'), composed of *min* ('poor') and *acht* ('respect'), versus underived *mi·na·ret* (id.). This is problematic since compounds, which are highly productive in Dutch, cannot be easily distinguished from underived words since they are written as one word. Hyphenation programs based on pattern matching, e.g. Boot (1984) and Tutelaers (1996), are essentially list-based as well: on the basis of a word list certain patterns are computed where hyphens can be inserted. Again, patterns which are correct in underived words may derive incorrect hyphens in derived words. Using a list-based program would considerably facilitate the task of hyphenating new words, but it would not

help to find rare errors and subtle inconsistencies in the words which are already hyphenated by hand.

In principle, an algorithm based on linguistic rules could be entirely correct. However, Daelemans reports an success rate of 99.88 %, even though Daelemans' algorithm does not provide all hyphenation positions (and although we will see below that at least some hyphens that Daelemans classifies as accurate are in fact incorrect). In Daelemans' view, hyphenation is based on (phonological) syllabification. Hyphens are therefore computed as follows: letters are converted to phonemes which are syllabified, and hyphens are inserted at syllable boundaries except in those places in which hyphenation is impossible, e.g. *taxi* (id.). This was in line with the literature on this subject at the time, cf. for instance Booij (1987), Wester (1985a/b), but more recent literature has drawn attention to the differences between phonological and orthographical syllables, cf. Nunn (1998). I will discuss these mismatches between phonological syllables and hyphenation in section 3. For this reason Daelemans' method in which hyphenation is essentially computed on the basis of phonological syllables seems less adequate.

Another drawback of Daelemans' approach is the fact that hyphenation is entangled with compound splitting. In some cases, compound boundaries can be predicted by phonotactic constraints; the occurrence of a consonant cluster which is not allowed morpheme-internally betrays the compound boundary. In *postzegel* ('stamp'), for instance, the sequence *stz* betrays the presence of the compound boundary between *t* and *z*. In other words compound boundaries cannot be predicted this way, cf. near minimal pairs such as *avon·tuur* ('adventure') versus *avond·uur* (*avond+uur*, 'evening hour'). In this case, Daelemans applies automatic compound analysis, but this is not flawless and does not offer a solution for ambiguous words since it does not involve semantic information. In other words, many errors reported by Daelemans are not hyphenation errors, but errors in the preceding morphological analysis. A more satisfactory hyphenation program should disentangle morphological analysis and hyphenation itself.

Evaluation of the hyphenation methods found in the literature shows that only rule-based hyphenation can eventually assign the correct hyphenation to all words, taking their morphological make-up into account. Furthermore, it can be successfully applied to new words, unlike methods based on word lists. However, the programs proposed so far fail to compute hyphenation by rules that are comparable to the hyphenation rules given by the orthography dictionary. Furthermore, to disentangle morphological analysis and hyphenation it is crucial that the input to the algorithm has already been morphologically analyzed.

3 Dutch hyphenation rules and their background

Hyphenation rules are given in *Woordenlijst van de Nederlandse taal*, the Dutch orthographic dictionary, henceforth denoted as [*Dutch word list 1995*]. The main rules are the following:

(1) Hyphens are inserted:

a) at compound boundaries: *min·achting* ('disdain'), after prefixes: *be·horen* ('to belong'), *her·ademen* ('breathe more freely'), before the suffixes *-aard* ('-ard') and *-achtig* ('-like'): *laf·aard* ('coward'), *waar·achtig* ('really'); and before suffixes beginning in a consonant: *boom·pje* ('little tree'), *dek·sel* ('lid'), etc.
b) between two adjacent vowels that do not denote one vowel like *eu*, *oe*, *ui*, *aai*, *ooi* and *oei*: *dooi·er* ('yolk'), *kri·oelen* ('to swarm')
c) after intervocalic *y:* *roy·aal* ('generous'), *relay·eren* ('to lead further')
d) before the maximal possible onset in intervocalic consonant clusters: *amb·ten* ('offices'), *art·sen* ('doctors'), *ek·ster* ('magpie'), *ern·stig* ('serious'), *erw·ten* ('peas'), *koort·sig* ('feverish'); *praktisch·te* ('most practical')
e) *st* and *sp* are split after the *s*: *oes·ter* ('oyster'), *has·pel* ('reel')
f) *ch* is one consonant: *bo·chel* ('bump'); *ng* consists of two consonants: *konin·gin* ('queen')
g) between two vowels there is no hyphenation before or after *x*: *exa·men* ('exam'), *exo·tisch* ('exotic')
h) in some cases hyphenated forms are written differently, e.g. *opaatje/opa·tje* ('little granddad')

These rules are subject to two additional conditions:

i) the position of hyphens may not suggest an incorrect pronunciation: *∗reg·lement* ('rules'), *∗qu·eue* (id.)
j) hyphenation may not leave a syllable of one separate letter at the end or beginning of a line. This also holds for words that are part of a compound or derivation: *∗a·drenaline* ('adrenalin'), *∗studi·o* (id.); *∗mensa·pen* (*mens+apen*, 'apes'), *∗vide·oachtig* (*video+achtig*, 'video-like').

At first sight these rules seem to be based on phonological syllables which in turn partly reflect morphological structure. On closer investigation, however, we see that this is not the case, as illustrated under (2) and (3). In the first place, although both phonological syllabification and hyphenation are sensitive to morphological structure, hyphenation reflects morphological structure in more cases, cf. the examples under (2), where ('@') denotes schwa. The first two examples show that hyphenation even disambiguates *bots+te* ('collided') and *bot+ste* ('rudest'):

(2)	*bots·te* (*bots+te*)	bOt-st@
	bot·ste (*bot+ste*)	bOt-st@
	waar·ach·tig	wa-rAx-t@x
	laf·aard	lA-fart
(3)	*oes·ter*	pri-st@r
	dooi·er	do-j@r
	taxi	tAk-si

The mismatches under (3) do not have a morphological background. For instance, the behavior of *oester* shows that the maximal onset principle is not always predominant in spelling. In the remaining two examples the choice of letters causes deviations between phonological and orthographical syllables, cf. also Nunn (1998). Note that words with (almost) the same phonological syllabification can show different hyphenation behavior dependent on their spelling, cf. *taxi* ([tAk-si]) vs. *ac·tie* ([Ak-si], 'action'), *dooi·er* ([do-j@r], 'yolk') vs. *go·jim* ([go-jIm], 'goyim'). It is not clear how Daelemans, who bases hyphenation on syllabification, handles these words, since he does not mention rules that adjust such mismatches between phonological and orthographical syllables (except in the case of *taxi*). This also raises questions about how he judged a given hyphen as correct or incorrect (especially since Daelemans also incorrectly assumes that *goochelaar* ('conjuror') is hyphenated as *∗go·che·laar* instead of *goo·che·laar*).

The facts under (2) and (3) may suggest that hyphenation is an autonomous process for which the pronunciation is irrelevant. This may be the case when a richer spelling representation is used which includes CV-structure, cf. Nunn (1998). However, since we can only refer to letter sequences, the pronunciation is crucial in some cases, e.g. *mu·s***e**·**u***m* (id.) vs. *kl***eu***m* ('frowster'), *op·ti·c***ie***n* ('optician') vs. *d***i**·**e***s* ('day'), *b***ea***t·nik* (id.) vs. *b***e**·**a**·*ti·fi·ca·tie* ('beatification').

Summarizing, to hyphenate words correctly it is necessary to have the correct spelling, the morphological make-up (i.e. markers that indicate the boundaries of prefixes, compound members and some suffixes) and the phonological representation of words.

4 Implementing and testing the rules

The starting point of the hyphenation program was formed by the hyphenation rules from [*Dutch Word List 1995*]. As discussed above, these rules often were not explicit enough, so many choices had to be made, for instance which nonnative morphemes were to be treated as compound members, and which clusters are allowed at the beginning of a syllable. Looking up relevant examples in the [*Dutch Word List 1995*] was no alternative, since the words were treated inconsistently there. The rules were written in such a way that words with variation in the pronunciation, which could possibly lead to hyphenation variation still get one possible hyphenation pattern only. For instance, *systeem* ('system') can be pronounced as [sistem] or [sIs-tem], but it must be hyphenated as follows: *sys.teem* An extra requirement was the following: since there is a condition that hyphenation positions should not suggest an incorrect pronunciation, we decided not to allow hyphens before mute vowels (*∗ra·ce*. [*Dutch Word*

List 1995] was not consistent in this respect, cf. *bl***ue***s* ([blu:z], id.) versus *g***u.e***rilla* ([G@-rIl-ja], 'guerrilla'). As a first step in the development of the ideal hyphenation rules, we decided to use rules that are based on spelling and morphological make-up only, and to leave the use of phonological representations aside for the moment. The rules were formalized by means of the computer language PERL. This language is suited for the formalization of linguistic rules, because of the use of regular expressions which facilitate string manipulation.

To ensure the consistent treatment of words and their inflected forms the hyphenation pattern of inflected words was derived in the following way: inflectional affixes are added to stems, and the spelling of the resulting word is computed by reapplying hyphenation at suffix boundaries only, while leaving the hyphenation in the rest of the word unaffected. This way, related words are treated consistently, while we allow for resyllabification at suffix boundaries, e.g. *leuk-leu·ke*. In the remainder of this paper, I will only discuss the computation of hyphens in uninflected words.

To be able to test the new algorithm, we need a set of hyphenated words of which the accuracy has already been established. Fortunately, almost all words in the dictionary database were already hyphenated. This was done by editors who applied the rules from [*Dutch Word List 1995*]. The editors did not insert hyphens (or omit them where hyphenation is impossible, e.g. in *taxi*), but used a more refined code. This is summed up in table 1. Normal hyphenation positions are not marked by hyphens, since this sign is also used to join words. Therefore the sign ('=') was chosen. When a hyphen coincides with a morphological boundary this is denoted by ('+') or ('@'). Exceptional hyphenation positions are marked with a ('!'). Note that all these signs are reduced to one notation '·' in the dictionary. Positions where hyphenation is not possible are further classified as ':' or '~'. Finally, a notation was introduced to encode the different spelling of the same word when it is or is not hyphenated. For instance, *o:paa[1|@]tje* should be interpreted as follows: when the word is not hyphenated the information within the brackets is ignored, but when it is hyphenated the information within the brackets means ('replace the last letter before the bracket by @'):

Table 1: Hyphenation symbols

Symbol	Explanation
=	syllable boundary, hyphenation possible
-	hyphen (the sign used to join two words) also syllable and compound boundary
+	word boundary, also syllable boundary
@	other morphological boundary which coincides with syllable boundary
:	syllable boundary after/before a single letter or before intervocalic *x*; hyphenation not permitted
~	unpredictable absence of a syllable boundary before mute vowels or within a digraph; hyphenation not permitted
!	marks unpredictable syllable boundaries
[|]	marks difference between hyphenated and unhyphenated variant

Table 2: Examples of the use of hyphenation symbols

Symbol	Example	Dictionary notation
=	*sche=mer*	*sche·mer*
-	*niet-ro=ker*	*niet-roker*
+	*min+acht*	*min·acht*
@	*boom@pje*	*boom·pje*
:	*a:vond*	*avond*
	ta:xi	*taxi*
~	*ra~ce*	*race*
	blu~es	*blues*
!	*mu=se=!um*	*mu·se·um*
[|]	*o:paa[1|@]tje*	*opaatje/opa·tje*

Because of this refined code, the hyphenated words in the database could be used as a test set for the new algorithm: by removing all hyphenation symbols except for the mor-

phological boundaries ('-'), ('+') and ('@'), we derive words provided with the morphological information necessary for the application of the hyphenation rules. Furthermore, ambiguous words were disambiguated, e.g. *be+ast* ('covered with ashes') versus *beast* (id.), *wets+taal* ('legal language versus *wet+staal* ('knife-sharpener'). Hyphenation rules were applied to these words, and the result was compared with the original set of hyphenated words. This way, it was possible to quickly detect errors in the implementation of the hyphenation rules.

5 Comparison of given and computed hyphenation positions

Even after all obvious errors of the rules had been corrected, there were still differences between the result of the hyphenation program and the words that were hyphenated by hand. The mismatches were examined and classified, and they turned out to fall into six classes:

Table 3: Mismatches between hyphenated words in the database and the result of the rules. The first hyphenated word is the word from the database; the word in parentheses is the form computed by the program; asterisks denote the incorrect forms:

1. errors	$*a \cdot \ddot{e} \cdot ro \cdot dy \cdot na \cdot misch$ ($a\ddot{e} \cdot ro \cdot dy \cdot na \cdot misch$ 'aerodynamic')
2. inconsistencies	$*trots \cdot kist$, $ra \cdot di \cdot ka \cdot lin \cdot ski$ ($trot \cdot skist$, 'Trotskyist', $ra \cdot di \cdot ka \cdot lin \cdot ski$, 'revolutionist')
3. variation	$*sy \cdot steem$, $sys \cdot teem$ ($sys \cdot teem$, 'system')
4. incorrect morphological analysis	$spel \cdot ling \cdot re \cdot gel$ ($*spel \cdot lin \cdot gre \cdot gel$, 'spelling rule')
5. errors due to the omission of the pronunciation	$race$ ($*rac \cdot e$, id.), $de \cdot us$ ($*deus$, 'god')
6. incorrect spelling rule	$co \cdot yo \cdot te$ ($*coy \cdot o \cdot te$, id.), $te \cdot ri \cdot ya \cdot ki$ ($*te \cdot riy \cdot a \cdot ki$, 'Japanese dish')

The first three types of mismatches could be attributed to flaws in the hyphenation positions that were added by hand. 1. gives an example of mere errors in the database. 2. illustrates identical letter sequences, e.g. a consonant followed by *sk*, which are treated inconsistently. In this case the refined hyphenation rules (*s*-clusters are parsed as onsets after consonant letters) generate a consistent pattern. 3. gives an example of variation in the database caused by variation in the pronunciation. The first vowel of the word *systeem* can be pronounced as a long [i] or a short [I], so the editors gave $sy \cdot steem$ as well as $sys \cdot teem$ as possibilities. However, since vowel length is irrelevant in native words (*st* is split after a short vowel in $bes \cdot te$ ('the best') as well as after a long vowel in $mees \cdot ter$ ('master'), only the second variant was allowed.

The remaining three types of mismatches had to be attributed to the hyphenation rules. In example 4., the incorrect result of the hyphenation rules is caused by the incorrect input: for instance, the compound *spelling+regel* ('spelling rule') in which the boundary between *spelling* and *regel* is not marked will be incorrectly treated as an underived word. Among this type of errors were also examples of nonnative words which had compound boundaries after morphemes we had decided not to treat as compound members, e.g. *an+algetisch* ('relieving pain') ($an \cdot al \cdot ge \cdot tisch$) instead of *analgetisch* ($anal \cdot ge \cdot tisch$) or vice versa: *Paleoceen* ($Pa \cdot le \cdot o \cdot ceen$) instead of *Paleo+ceen* ('Palaeocene') ($Pa \cdot leo \cdot ceen$). The type of error illustrated by 5. was unavoidable since we did not yet take the pronunciation into account. For this reason, hyphens are incorrectly inserted in words with mute vowels (*race* ([res]), and incorrectly omitted in words where vowel sequences that normally encode one vowel, represent two sounds and where this special spelling is not marked by dieresis, e.g. $de \cdot us$ ([de-jUs], 'god'). 6. illustrates an interesting type of error: even though rule (1c) from [*Dutch Word List 1995*], repeated below as (4) was formalized accurately, the hyphenation computed by the rules seemed counterintuitive in words such as $*coy \cdot o.te$, $*te \cdot riy \cdot a.ki$.

(4) hyphens are inserted after intervocalic *y*: $roy \cdot aal$, $relay \cdot eren$

It seemed that in this case the rule is incor-

rect. This was supported by the fact that the formulation of the same rule was subtly, but also crucially different in 1954:

y in words such as *royaal*, *relayeren* is part of the first syllable (*Dutch Word List 1954*, p. LIII).

y is part of the first syllable: *loyaal*, *relayeren* (*Dutch Word List 1995*, p. LIII).

In other words, *royaal*, *relayeren* are not just examples of the rule but they illustrate a restriction. In both these words *y* and the preceding vowel form a digraph; or at least they used to form a digraph in 1954, but in *coy·o.te* and *te·riy·a.ki* they do not. *y* is only part of the first syllable when it is part of digraph.

The mismatches were removed in the following way: the errors in the database (1.-4.) were corrected. The errors under 5. could not yet be solved, so the relevant words were marked as exceptions for the time being. Finally, the inaccurate rule for the hyphenation of intervocalic *y* of [*Dutch Word List 1995*] was replaced by the more accurate version of 1954.

This implies that when the hyphenation rules are applied to new unhyphenated words, hyphens will be inserted correctly, except in a few foreign words such as *race* and *deus*.

6 Conclusion

We developed a hyphenation program to improve the quality of a dictionary database, and to provide new words with hyphenation positions. Earlier hyphenation programs combine hyphenation with morphological analysis. Since this is not flawless, the potential accuracy of hyphenation rules is underestimated. The program described here has input data which are provided with all relevant lexicographic information such as morphological make-up and pronunciation, so it is in principle possible to correctly predict hyphens in all words.

However, since the pronunciation is not yet taken into account, a number of words had to be marked as exceptions. This also implies that the program will predict incorrect hyphenation patterns in some new foreign words. However, we intend to add a second step to the hyphenation program which will remedy this shortcoming by using phonological information. After this adjustment, we expect the program to perform better than all previous algorithms.

By using the hyphenation program to recompute the hyphens of words already hyphenated, it is possible to identify errors and inconsistencies in the database. Interestingly, the comparison also revealed an incorrect formulation of one of the official hyphenation rules that has been introduced with the Dutch spelling reform of 1995. These results could not have been achieved with previous methods based on word lists or phonological syllables.

These promising results suggest that recomputing data on the basis of linguistic rules can also improve the consistency of other parts of the database. For example, grapheme-to-phoneme conversion rules could be used to increase the consistency of the phonological representations in the database.

References

Booij, G.E.(1987), The Reflection of Linguistic Structure in Dutch Spelling, *in* P.A. Luelsdorff (ed.), *Orthography and Phonology*, John Benjamins, Amsterdam-Philadelphia, pp. 215-224.

Boot, M.(1984), *Taal, tekst, computer*, Servire, Katwijk.

Brandt Corstius, H.(1970), *Exercises in Computational Linguistics*, Mathematical Centre Tracts 30, Amsterdam.

Daelemans, W.(1989), Automatic hyphenation: linguistics versus engineering, *in* F. J. Heyvaert and F. Steurs (eds.), *Worlds behind words*, Leuven University Press, Leuven, pp. 347-364.

Daelemans, W.(1987), *Studies in Language Technology. An Object-Oriented Computer Model of Morphophonological Aspects of Dutch*, PhD thesis, University of Leuven.

Geerts, G. and T. den Boon (1999), *Van Dale groot woordenboek der Nederlandse taal. Dertiende, herzien uitgave*, Van Dale lexicografie Utrecht-Antwerpen.

Liang, M.(1983), *Word hy-phen-a-tion by Comput-er*, PhD thesis, Stanford University.

Nunn, A.M.(1998), *Dutch Orthography. A Systematic Investigation of the Spelling of Dutch Words*, Holland Academic Graphics, Den Haag.

Tutelaers, P.(1993), Herziene afbreekpatronen voor het Nederlands, *MAPS* 1993, pp. 187-190.

Wester, J.(1985a), Language Technology as Linguistics: A Phonological Case Study of Dutch Spelling, *in* H. Bennis and A. van Kemenade (eds.), *Linguistics in the Netherlands 1985*, Foris Dordrecht, pp. 205-212.

Wester, J.(1985b), Autonome Spelling en Toegepaste Fonologie, of: naar een generatieve spellingtheorie, *Gramma* 9, pp. 173-196.

[*Dutch Word List 1954*], *Woordenlijst van de Nederlandse Taal, samengesteld in opdracht van de Nederlandse en de Belgische regering*, SDU uitgeverij, Den Haag, 1954.

[*Dutch Word List 1995*], *Woordenlijst Nederlandse taal. Samengesteld door het Instituut voor Nederlandse lexicologie in opdracht van de Nederlandse Taalunie*, SDU Uitgevers/Standaard Uitgeverij, Den Haag-Antwerpen, 1995.