

# A syllable frequency list for Dutch

Willem Zuidema

December 31, 2009

## 1 Introduction

The Corpus Gesproken Nederlands (CGN) is a large corpus of spoken Dutch, partly annotated with syntactic and phonological information (see <http://lands.let.kun.nl/cgn/>). Although it contains files with syllabified words, and word frequency counts, there is no direct way to extract from it a list of syllable frequencies. This document describes some simple scripts to combine the relevant information from various CGN files (using version 6 and the linux utilities `grep`, `sed`, `sort`, `uniq`, `awk`, `cut` and `paste`), and gives a complete list of syllable frequencies obtained by running the scripts. The list is made available in the hope that it might be helpful, for instance for experimental studies where one must control for syllable frequency. Depending on the intended use or required level of accuracy, the scripts might have to be adapted and the frequency counts changed accordingly.

## 2 Syllable frequencies

To determine syllable frequency, we combine information from two sources: word frequencies as provided by the CGN (presumably based on the orthographic form of words in the transcriptions) and word syllabification, also provided by CGN (presumably based on native-speaker intuitions about words considered in isolation). We thus make the important assumptions that syllable frequencies are simply the sum of frequencies of the words in which they occur, and that words have a single syllable-structure that is independent from the context.

CGN's word frequency table has 143850 entries (representing counts of a total of 6153974 word tokens), of which 22623 entries (15.7% type frequency; representing 108889 tokens or 1.8% token frequency) are specially marked as 'incomplete' (5664), 'mispr' (5483), 'foreign' (5028), 'uncertain' (4224), 'dialect' (1401) or 'regionalpr' (823). For further calculations, these entries are ignored, but for some purposes the syllable frequencies might have to be adjusted to take also these words into account. The following table gives the 40 most frequent of such ignored words:

Table 1: Top 40 ignored words

Frequency	Word/annotation
3158	d/incomplete
2553	ne/dialect
2340	de/dialect
1873	m/incomplete
1791	i/incomplete
1664	den/dialect
1604	v/incomplete
1449	w/incomplete
1407	n/incomplete

Continued on next page

Table 1 – continued from previous page

Frequency	Word/annotation
1359	da/incomplete
1284	s/incomplete
1142	z/incomplete
1074	'k/dialect
1056	o/incomplete
1018	nen/dialect
963	a/incomplete
906	wa/incomplete
867	g/incomplete
856	e/incomplete
789	ge/incomplete
784	ja/uncertain
756	t/incomplete
718	dat/uncertain
703	b/incomplete
655	j/incomplete
630	k/incomplete
532	maar/regionalpr
495	dan/uncertain
474	he/incomplete
450	diejen/dialect
426	een/uncertain
412	dieje/dialect
399	ma/incomplete
398	en/uncertain
396	be/incomplete
386	't/uncertain
386	maar/uncertain
380	me/incomplete
367	we/incomplete
360	de/uncertain

Technically, to combine the information from the two CGN-files we go through the following steps: First, we create a file `word-syllable.txt` with words and their syllabified versions according to the lexicon file provided with CGN. That is, we copy `cnlex6.txt` and remove all but the appropriate columns (labeled 'Orthography' and 'Uitspraak CELEX' in the corresponding html file), and reverse the order. The file starts with (the three occurrences of 'en differ in other attributes in the lexicon file, but only the first occurrence will be used later on):

```
'en &eacute;&eacute;n
'en &eacute;&eacute;n
'en &eacute;&eacute;n
y-b@r-'hA+pt &uuml;berhaupt
's 's
```

Next, we create a file `word-freq.txt` with words and their frequencies. That is, we copy the file `totrank.frq` and remove all but the column 'TOT' and 'TOKEN'. The file starts with:

```
193713 de
169559 dat
```

```
163688 ja
149072 en
127469 uh
```

Then we sort the files on word tokens, and merge the corresponding entries from the two files:

```
sort -k 2 -u word-syllable.txt > ws.txt
sort -k 2 -u word-freq.txt > wf.txt
join -j 2 wf.txt ws.txt > wsf.txt
```

Finally, we put every syllable with the frequency from the word it comes from, on a separate line, remove stress annotation, sort the output so that observations of the same syllable occur next to each other, and sum the counts. (Because the `sed` command only applies once to a line, it is repeated here as often as the highest number of syllables per word observed. Not the prettiest solution, but it works).

```
cat wsf.txt
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| sed 's/\(.+\ [0-9]\+\) \([^-]\+\)-\(.+\)/\1 \2\n\1 \3/'
| cut -f2-3 -d ' ' > syllperline.txt
cat syllperline.txt | sed "s/'//g" | sort -k 2 > spl2.txt
cat spl2.txt | awk 'BEGIN {prevc=0;prevw="nil";s=0}; {if ($2==prevw)
prevc+=$1; else {print prevc "\t" prevw; prevc=$1; prevw=$2}}
END {print prevc "\t" prevw; prevc=$1}' > syllable-frequencylist6.txt
```

To order the results on frequency, we just run:

```
cat syllable-frequencylist6.txt | sort -g -r -k 1 > syllable-frequencylist7.txt
```

The result is a list of 7087 syllables (some resulting from annotation errors). The top 20 is below (with CV-type annotation as explained in next section); see the appendix (available online at <http://staff.science.uva.nl/~jzuidema/research/>) for the complete list.

Table 2: Syllable frequencies

Frequency	Syllable	Type
171287	ja	CV
135261	en	VC
129175	x@	CV
127538	d@	CV
116746	Ik	VC
103258	t@	CV
92983	Is	VC
85706	G@	CV
Continued on next page		

Table 2 – continued from previous page

Frequency	Syllable	Type
78262	b@	CV
72925	nit	CVC
66299	dAn	CVC
65055	k@	CV
53008	l@	CV
50100	v@r	CVC
49224	r@	CV
48731	vor	CVC
44665	hE	CV
44581	ne	CV
44349	wEl	CVC
44262	l@k	CVC

### 3 Syllable types

To count the number of syllable types (CVC etc), we split the frequency list in two files, one with frequencies, the other with the syllables. We then replace all consonants (defined as elements from the set {pbtldkgfvszSZxGhNmnJlrwj}) with C and all vowels (defined as all other symbols optionally followed by a sequence from the set {:+}) with V in the latter. Next, we rejoin the files and/or sum frequencies of occurrences of the same {CV}\* string.

```
cut -f 1 syllable-frequencylist7.txt > /tmp/freq
cut -f 2 syllable-frequencylist7.txt |
  sed 's/[pbtldkgfvszSZxGhNmnJlrwj]/C/g;s/[^C][:+]*/V/g' > /tmp/CV
paste /tmp/freq /tmp/CV | sort -k 2 |
  awk 'BEGIN {prevc=0;prevw="nil";s=0}; {if ($2==prevw) prevc+=$1;
  else {print prevc "\t" prevw; prevc=$1; prevw=$2}} END {print
  prevc "\t" prevw; prevc=$1}' |
  sort -g -r -k 1 > CV-freqlist2.txt
paste syllable-frequencylist7.txt /tmp/CV > syllable-frequencylist8.txt
```

The result are the following frequencies of syllable types:

Table 3: Syllable type frequencies

Frequency	Type
1931788	CV
1584685	CVC
518071	VC
340217	CVCC
192292	CCV
156922	CCVC
95632	V
55661	CCVCC
51902	VCC
18751	CVCCC
13763	C
Continued on next page	

Table 3 – continued from previous page

Frequency	Type
12603	CCCV
9420	CCVC
2895	CCVCC
1666	CCVCCC
1050	VCCC
148	CVCCCC
36	CCCVCCC
34	VCCCC
11	CC
9	CCVCCCC
2	CCC
4987558	Total

## 4 Phoneme frequencies

First we put all phonetically transcribed sentences in one file `cgn-fonetisch.txt`.

```
less r*/fon/fn* | grep -E '^"[^"]' | grep -v -E
'(IntervalTier)|(UNKNOWNN)|(TextGrid)|(N.....-fontr)' >
/scratch/jzuidema/corpora/cgn-fonetisch.txt
```

Then we put all occurring phonemes on a unique line, sort and count.

```
cat ~/scratch/corpora/cgn-fonetisch.txt | sed 's/[ ]//g' | sed
's/-.[+~:~:~]-//g' | sed 's/(\.[+~:~:~]*\)/\1\n/g' | grep -v -E '^$' |
grep -v -E '[]\[\*_~]' | sort | uniq -c | sort -g -r -k 1 >
phoneme-freq.txt
```

Note that the regular expressions are used here to recognize phonemes that are represented with several characters in the CELEX notation, following these conventions:

- +~: are used to mark diphthongs etc
- \_ to mark a silent phoneme (only there to keep correspondence with orthography; to be ignored)
- to mark an inserted phoneme (not in the orthography; to be counted) (not all hyphens are in pairs as they should be according to manual)
- [] mark unknown sounds (to be ignored)
- \*.^ signs and several numbers (other than 2) are not in the manual

The result is the following list of phoneme frequencies against rank (graphically represented in figure 2):

Table 4: Phoneme frequencies

Frequency	Phoneme
179691	@
114069	t
110814	n
80134	d
Continued on next page	

Table 4 – continued from previous page

Frequency	Phoneme
77391	r
66913	s
66889	A
54275	l
48432	a
47854	k
47200	E
46505	m
41199	I
39540	e
39447	x
33541	o
33252	i
33228	O
32593	w
31693	f
23806	j
23122	h
22932	b
21527	p
21492	E+
20071	z
17367	v
13846	N
13222	u
12824	G
9505	Y
6659	y
5888	A+
5640	Y+
4881	g
2463	2
2411	#
1970	S
1215	0
996	J
481	Z
378	U
378	K
370	W
215	E:
188	1
140	3
94	5
87	O:
87	4
70	8
69	A~

Continued on next page

Table 4 – continued from previous page

Frequency	Phoneme
64	7
42	6
33	9
32	E~
31	Y:
16	M
16	C
15	O~
8	T
8	R
8	D
8	B
2	Y~
2	.
1	V

## 5 Zipf curves

Plotting the obtained frequencies of words, syllables, phonemes and POS-tags against their rank in their respective frequency lists, we see that only the last curve follows Zipf’s law, while the others bend downwards towards the right of the graph (the POS-tags curve turns first, then phonemes, then syllables).

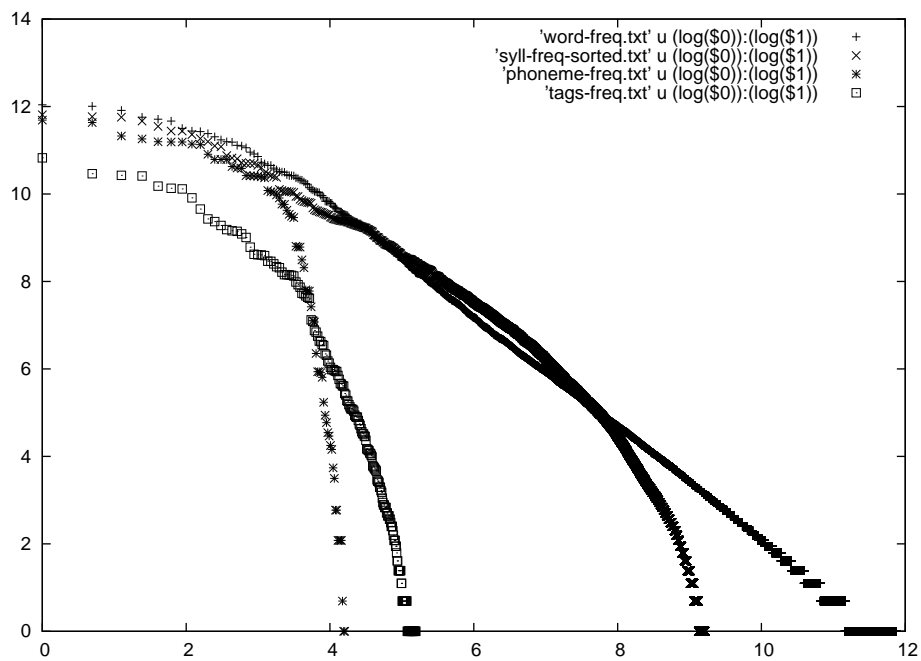


Figure 1: Frequency against rank in a log-log space, for Dutch word, syllable, phoneme and POS-tag distributions according to the CGN.

**Acknowledgments** This work was carried out in part following the suggestions made by Griet Depoorter to Sita ter Haar.



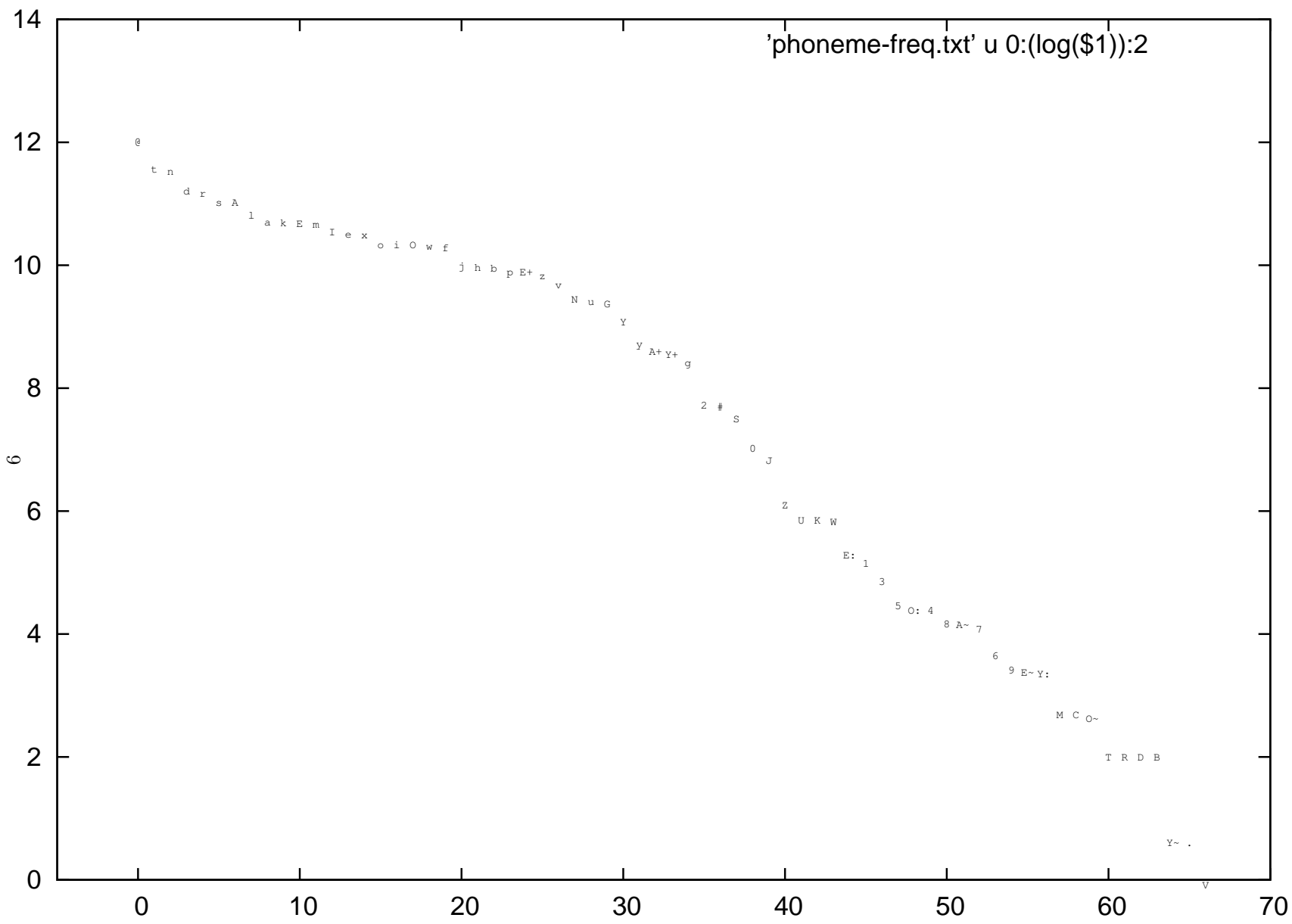


Figure 2: Phoneme (log) frequencies against rank; the frequency distribution appears to be following an exponential distribution (showing a more or less straight line in a log-linear space).