Spoken Language Generation
Project "het spreken van cijfers"

Marijn Koolen (9900519)
Frans Adriaans (9930787)
Ork de Rooij (9901655)

4 juni 2004

Contents

1 Inleiding

Dit document beschrijft ons projectvoorstel voor het vak Spoken Language Generation. Ons voorstel is om alle denk- en processtappen te doorlopen die nodig zijn om een Nederlandse TTS voice te maken voor het uitspreken van cijfers.

Wij hebben gekozen voor het uitspreken van cijfers, omdat volgens ons dit een mooie, afgeronde, subset geeft. Als we rekening houden met slechts de cijfers 0 t/m 9, en in eerste instanties geen concatenaties als "eenentwintig" toestaan dan is de set (di)fonen die benodigd zijn om dit uit te kunnen spreken precies bekend. Hierdoor is het mogelijk om een werkende synthesizer stem te krijgen binnen een beperkte tijd, en tijdens dit proces een duidelijk 'gevoel' te krijgen over hoe het maken van een stem precies werkt, en welke problemen hierbij komen kijken. Dit allemaal zonder al te veel te vervallen in het behandelen van uitzonderingen en het eeuwig blijven toevoegen van weinig voorkomende difooncombinaties.

Ons idee is om ook zelf een stem op te nemen, deze zelf te segmenteren, en met gebruik van bestaande tools te integreren in festival. Het benodigde lexicon, het intonatiemodel en het durationmodel willen we hiervoor zelf opstellen. Het uiteindelijke doel is dat festival met een commando als `(SayText "42")` uit de voeten kan en dit laat uitspreken met de door ons gegenereerde stem.

In dit document beschrijven wij welke stappen er, volgens ons, nodig zijn om deze stem te maken, en wat voor problemen wij denken hierbij tegen te komen.

2 De stappen

Wij hebben de volgende stappen onderscheiden in het proces om een stem te maken. Als leiddraad hiervoor hebben wij het hoofdstuk "Building a new voice" uit de festival manual[1] gebruikt.

2.1 Het bepalen van welke cijfers we willen kunnen uitspreken

Zoals genoemd in de inleiding willen we ons in eerste instantie beperken tot het uitspreken van de cijfers 0 t/m 9 Om cijfers uit te kunnen spreken hebben we regels nodig die de cijfers

[1] URL `http://www.cstr.ed.ac.uk/projects/festival`

omzetten in hun woorden. Omdat het uitspreken van alle cijfers een te grote hoeveelheid werk met zich mee zal brengen

2.2 Het schrijven van een lexicon: welke phones hebben we nodig

De cijfers, of eigenlijk de woorden "een" tot en met "negen" bevatten een aantal verschillende fonen. De eerste stap die we gaan maken is het opschrijven van deze woorden in hun fonetisch schrift, om vervolgens de benodigde difoonset hieruit af te leiden. Van de lijst fonen wordt een lijst van difonen gemaakt. Voor het woord 'een', bestaande uit de fonen *e:* en *n*, zijn de volgende difonen nodig: `#-e:`, `e-:n`, en `n-#`. Elk difoon bestaat uit de laatste helft van de eerste foon en de eerste helft van de tweede foon. Aan het begin en het eind van elk woord is een overgang naar het stilte foneem (#) nodig. In appendix A staat een volledige lijst met alle difonen. In appendix B staat het uiteindelijke lexicon.

2.3 Het opnemen van een stem

2.3.1 Het voorbereiden van de opname

Er is een mini-corpus opgenomen om de benodigde geluidsfragmenten te verkrijgen. Hieruit is vervolgens de difoonset ge-extraheerd. Deze lijst is te vinden in appendix A. Om een natuurlijke uitspraak te krijgen, hebben we de woorden uit ons lexicon geplaatst in een context. Als context is de volgende woordcombinatie gekozen:

```
aap nul pak
aap een pak
 ...
aap negen pak
```

Er is bewust gekozen voor de plaatsing van het foneem *'p'* vlak voor en vlak na het lexiconwoord. Het voordeel van de plaatsing van dit plosief vlak na het lexiconwoord, is dat het de spreker dwingt het stilte-foneem uit te spreken. Tussen het eerste woord en het lexiconwoord is bewust een kleine rust ingelast, om ook hier het stiltefoneem te krijgen. Op deze manier wordt tevens voorkomen dat de eerste en laatste fonemen opgaan in de omliggende woorden.

2.3.2 Het opnemen

Bij het maken van de opnames zijn niet de meest stricte eisen gesteld aan de omstandigheden. Dit heeft direct negatieve invloed op de uiteindelijk synthese-kwaliteit. De stem van Frans is vroeg in de middag opgenomen in een redelijk stille kamer. De gebruikte microfoon is echter van matige kwaliteit. Bij een volgende sessie zou geprobeerd moeten worden een betere microfoon te regelen, met bijvoorbeeld een filter ervoor, om te voorkomen dat het foneem *'p'* niet zo snel overstuurt. Ook zou een meer geoefende spreker met een heldere

stem bijdragen aan een beter resultaat. De opnames zijn gemaakt met behulp van het programma Praat. Van elke zin zijn een aantal opnames gemaakt, waarvan de beste versies uiteindelijk zijn gebruikt voor het extraheren van de difonen. Getracht is om de zinnen zo monotoon mogelijk uit te spreken. Ook is geprobeerd de afstand tussen spreker en microfoon constant te houden, opdat de uiteindelijke geluidsfragmenten niet te verschillen qua frequentiebereik en qua hoeveelheid ruis.

2.3.3 Het segmenteren naar difonen

Met behulp van Praat zijn de lexiconwoorden geanalyseerd. Per woord zijn eerst alle foneemgrenzen bepaald. Vervolgens zijn difonen geextraheerd, die van de helft van het ene foon tot de helft van het volgende foon lopen. Voor de start- en einddifonen is steeds een stuk stilte meegenomen die ongeveer even lang is als de helft van die start- of eindfoon. Elk difoon is voor verdere bewerking opgeslagen als WAV bestand. De uiteindelijke difoonset bestaat uit 37 WAV bestanden. Zie appendix B voor meer informatie over wat in welke file staat.

3 Het maken van de stem

Wij gebruiken de festvox toolkit[2] om de stem te maken. Deze toolkit bevat diverse scripts en tools om van de opgenomen stemfragmenten een festival voice te maken.

Als voice synthesizer hebben wij gekozen voor de in festival aanwezige UniSyn LPC diphone synthesizer. Dit omdat:

- De festvox handleiding een stap-voor-stap uitleg geeft over hoe een nieuwe stem met deze synthesizer gemaakt kan worden...
- we hierdoor zeer veel controle hadden over elke stap...
- ... bijna alles zelf konden aanpassen ...
- ... en er dus ook veel van konden leren door te experimenteren.
- Het maken van een MBROLA stem niet realiseerbaar was binnen de tijd: hiervoor zou MBROLA mee moeten werken om de difoonsynthesizer te genereren.

3.1 Aanmaken van een raamwerk voor de nieuwe stem

Met het festvox script `setup_diphone` hebben wij een festival voice directorystructuur aangemaakt. Hierdoor zijn alle benodigde directories en scripts aanwezig, en kan festival practisch gelijk gestart worden:

```
> $FESTVOXDIR/src/diphones/setup_diphone net dutch frans
```

[2] www.festvox.org

In deze directorystructuur hebben wij vervolgens de geluiden naar de `wav/` directory gekopieerd.

3.2 Labellen

Om de festvox toolkit een difoonstem te kunnen laten maken was het nodig om de gegenereerde WAV files te labellen, d.w.z. om aan te geven tussen waar en waar het difoon voorkwam in het geluidsfragment. In eerste instantie was gedacht dat dit triviaal zou zijn: de difoonsegmenten waren al keurig opgedeeld, en het labellen kon dus gebeuren door 1 label aan het begin, en 1 aan het einde van elk WAV bestand te plaatsen. Tijdens het generatieproces bleek echter dat dit niet goed klonk, en daarom zijn alle difonen alsnog handmatig gelabeld.

Het labellen van de losse difonen is gedaan met het programma `wavesurfer 1.6.4`[3], omdat deze het benodigde label formaat gelijk kon maken. De resulterende labels zijn opgeslagen in `lab/`

3.3 Het generatieproces

Het was noodzakelijk om het de stem meerdere malen te hergenereren, en na elke itteratie te luisteren welke cijfers goed uitgesproken werden, en welke niet. Om dit te vergemakkelijken is het script `compile` geschreven, wat alle stappen noodzakelijk om de stem gebruiksklaar in festival te krijgen doorloopt. Compile doet alle stappen zoveel mogelijk batchgewijs en automatisch.

Een itteratie ziet er ongeveer zo uit:

1. "tweaken": wijzigen van geluiden en labels met `wavesurfer`. Eventuele aanpassingen in `etc/diphones.list`, het lexicon, het F0 model, de foneemset en het duratie model.
2. `./compile`
3. `./run`
4. Beluister de resultaten
5. Is alles goed? Klaar. Anders: ga naar 1

3.4 Ontstane problemen

- We kwamen er al snel achter dat we geen stile-foneem hadden (dwz. `#-#`). Deze hebben we toegevoegd door een stukje stilte te labellen als `#-#`.
- Sommige diphones werden niet gevonden door festival, bijvoorbeeld bij het woord "drie". De melding `"UniSyn: using default diphone #-# for d-r"` wordt dan gegeven, en het resulterende geluid bevat ook geen "d-r" klank. Dit heb ik opgelost

[3] http://www.speech.kth.se/wavesurfer/

door de r foneem te hernoemen in R, waarna het wel werkt. Waarom dit probleem optreed is me een raadsel.

- In eerste instantie deed de F0 synthesizer het niet: alles werd zeer monotoon uitgesproken. Door accenten te zetten op alle woorden (of in het geval van "zeven" en "negen" alleen op de klemtoon) werd dit verholpen.

- Veel delen van festival, festvox en speech_tools weigeren te compileren onder GCC 3.3.2. Om dit op te lossen moesten er een flink aantal patches worden geschreven.

4 Festival code

Om de stem vervolgens ook echt te laten werken moesten er nog een aantal scheme files geschreven worden. Dankzij de festvox toolkit was hiervoor al een mooi raamwerk beschikbaar in de `festvox/` subdirectory, maar hier wordt dan wel gebruik gemaakt van een standaard F0, en is er nog geen duratie model. Ook het lexicon en de beschikbare fonemen moeten zelf worden toegevoegd. En voor het uitspreken van cijfers was het van belang een tokenizer te schrijven die cijfers omzette naar in het lexicon aanwezige woorden. In appendix B staan de gemaakte wijzigingen uitgelegd, samen met de geschreven sourcecode.

5 Evaluatie en conclusie

Het beoogde doel: het vanaf de grond opbouwen van een stem voor Festival is bereikt. Het resultaat klinkt nog ergens naar ook in onze oren, maar dat kan komen doordat wij het teveel hebben moeten horen. De kwaliteit is echter bij lange na niet dezelfde als die van een "echte" festival voice. Dit was echter ook niet ons ambitieniveau, dus dat maakt niet uit. Wij hebben van het proces echter wel veel kunnen leren.

5.1 Toekomst

We hebben de gemaakte festival voice nog niet ge-exporteerd voor distributie. Dit vanwege (vermoedelijke) copyright issues: we hebben nergens copyright bijgezet, en ons ook niet uitvoerig verdiept in welke onderdelen we waarvandaan hebben gebruikt. Als test hebben we wel een distributie versie gemaakt aan de hand van de beschrijving in de festvox documentatie (halverwege hoofdstuk 19).

Met het commando `(us_make_group_file "group/franslpc.group" nil)` kan een group file aangemaakt worden met daarin alleen de difoongeluidsfragmenten. Deze file is, samen met de schema code in de `festvox` directory voldoende om de stem ook op andere computers te installeren. Documentatie hierover staat in de festvox handleiding.

A Difoonlijst

Hieronder staat de lijst met fonemen en difonen die nodig zijn voor het uitspreken van de cijfers 0 t/m 9.

Woord	Fonemen	Difonen
nul	*n ux l*	`#-n n-ux ux-l l-#`
een	*e: n*	`#-e: e:-n n-#`
twee	*[V e:*	`#-t t-V[ V[-e: e:-#`
drie	*d r i:*	`#-d d-r r-i: i:-#`
vier	*v i: r*	`#-v v-i: i:-r r-#`
vijf	*v Ei f*	`#-v v-Ei Ei-f f-#`
zes	*z e s*	`#-z z-e e-s s-#`
zeven	*z e: v & n*	`#-z z-e: e:-v v-& &-n n-#`
acht	*a x t*	`#-a a-x x-t t-#`
negen	*n e: G & n*	`#-n n-e: e:-G G-& &-n n-#`

B Geschreven files

B.1 compile

```
#!/bin/bash
#
# simple 'compile' script to create the net_nl_frans_diphone voice
# from the wavs and the labels.
#
# (c) 2004 Ork de Rooij
#     email: ork@ork.be

export FESTVOXDIR=/home/ork/src/festival/festvox
export ESTDIR=/home/ork/src/festival/speech_tools
export PATH=$PATH:/home/ork/src/festival/speech_tools/bin:
export LD_LIBRARY_PATH=/home/ork/src/festival/speech_tools/lib

echo "Moving residual labels to the right directory..."
cp wav/*.lab lab >/dev/null

echo "Creating phoneme dictionary..."
bin/make_diph_index etc/diphones.list dic/fransdiph.est >/dev/null

echo "Creating pitchmarks..."
bin/make_pm_wave wav/*.wav >/dev/null

echo "Correcting pitchmarks..."
```

```
bin/make_pm_fix pm/*.pm >/dev/null

echo "Finding powerfactors..."
bin/find_powerfactors lab/*.lab >/dev/null

echo "Calculating LPC coefficients..."
bin/make_lpc wav/*.wav >/dev/null

echo ".. done."
```

B.2 etc/diphones.list

In deze file staan verwijzingen naar alle .wav bestanden, en welke difonen hier in voorkomen. Dit bestand wordt gebruikt door `bin/make_diph_index`. Bestanden 1 t/m 7 zijn de begindifonen, bestanden 8 t/m 29 zijn de tussendifonen, en bestanden 30 t/m 37 zijn de einddifonen.

```
( 1  "#-n"    ( "#-n"  ) )
( 2  "#-e:"   ( "#-e:" ) )
( 3  "#-t"    ( "#-t"  ) )
( 4  "#-d"    ( "#-d"  ) )
( 5  "#-v"    ( "#-v"  ) )
( 6  "#-z"    ( "#-z"  ) )
( 7  "#-a"    ( "#-a"  ) )
( 8  "n-ux"   ( "n-ux" ) )
( 9  "ux-l"   ( "ux-l" ) )
( 10 "e:-n"   ( "e:-n" ) )
( 11 "t-V["   ( "t-V[" ) )
( 12 "V[-e:"  ( "V[-e:") )
( 13 "d-R"    ( "d-R"  ) )
( 14 "R-i:"   ( "R-i:" ) )
( 15 "v-i:"   ( "v-i:" ) )
( 16 "i:-R"   ( "i:-R" ) )
( 17 "v-Ei"   ( "v-Ei" ) )
( 18 "Ei-f"   ( "Ei-f" ) )
( 19 "z-e"    ( "z-e"  ) )
( 20 "e-s"    ( "e-s"  ) )
( 21 "z-e:"   ( "z-e:" ) )
( 22 "e:-v"   ( "e:-v" ) )
( 23 "v-&"    ( "v-&"  ) )
( 24 "&-n"    ( "&-n"  ) )
( 25 "a-x"    ( "a-x"  ) )
( 26 "x-t"    ( "x-t"  ) )
```

```
( 27 "n-e:"   ( "n-e:" ) )
( 28 "e:-G"   ( "e:-G" ) )
( 29 "G-&"    ( "G-&"  ) )
( 30 "l-#"    ( "l-#"  "#-#" ) )
( 31 "n-#"    ( "n-#"  ) )
( 32 "e:-#"   ( "e:-#" ) )
( 33 "i:-#"   ( "i:-#" ) )
( 34 "R-#"    ( "R-#"  ) )
( 35 "f-#"    ( "f-#"  ) )
( 36 "s-#"    ( "s-#"  ) )
( 37 "t-#"    ( "t-#"  ) )
```

B.3 festvox/net_dutch_frans_phoneset.scm

Dit is de geschreven foneemset. Als features hebben wij de nextens featurelijst gebruikt, maar deze worden vervolgens zelf niet meer ergens gebruikt. Hieronder is alleen het belangrijke deel vermeld:

```
\begin{verbatim}

(defPhoneSet
  net_dutch
  ;;;  Phone Features
  (;; vowel or consonant
   (vc + - 0)
   ;; vowel length: short or long
   (vlng s l 0)
   ;; vowel type: short long dipthong schwa
   (vtype s l d a 0)
   ;; vowel height: high mid low
   (vheight 1 2 3 0)
   ;; vowel frontness: front mid back
   (vfront 1 2 3 0)
   ;; lip rounding
   (vrnd + - 0)
   ;; consonant type: plosive fricative affricative nasal liquid
   (ctype p f a n l r 0)
   ;; place of articulation: labial alveolar palatal labio-dental
   ;;                        dental velar glottal
   (cplace l a p b d v g 0)
   ;; consonant voicing
   (cvox + - 0)
```

```
   )
  (
     (#   0   0   0   0   0   -   0   0   -) ;; silence

     (ux  +   s   s   2   3   +   0   0   0) ;; nUl
     (e:  +   l   l   2   1   -   0   0   0) ;; EEn
     (a   +   s   s   3   3   -   0   0   0) ;; Acht
     (i:  +   s   s   1   1   -   0   0   0) ;; drIE
     (Ei  +   l   d   3   2   -   0   0   0) ;; vIJf
     (e   +   l   l   2   1   -   0   0   0) ;; zEs
     (&   +   s   s   2   3   +   0   0   0) ;; negEn

     (n   -   0   0   0   0   0   n   a   +) ;; Nul
     (V[  -   0   0   0   0   0   l   l   +) ;; tWee
     (t   -   0   0   0   0   0   p   a   -) ;; Twee
     (d   -   0   0   0   0   0   p   a   +) ;; Drie
     (R   -   0   0   0   0   0   r   a   +) ;; dRie
     (l   -   0   0   0   0   0   l   a   +) ;; nuL
     (s   -   0   0   0   0   0   f   a   -) ;; zeS
     (x   -   0   0   0   0   0   f   g   -) ;; aCHt
     (G   -   0   0   0   0   0   f   g   +) ;; neGen
     (f   -   0   0   0   0   0   f   b   -) ;; vijF
     (v   -   0   0   0   0   0   f   b   +) ;; Vijf
     (z   -   0   0   0   0   0   f   a   +) ;; Zes
  )
)

(PhoneSet.silences '(#))
```

B.4 festvox/net_dutch_frans_lexicon.scm

Omdat dit project toch over slechts een zeer beperkte woordenschat gaat hebben wij gewoon alle woorden in het lexicon gestopt. Het lexicon bevat verder dus ook geen letter to sound rules voor als er een onbekend woord langskomt. Wel is er op elk woord een accent gezet, omdat anders de F0 synthesizer niet natuurlijk klonk. Hieronder is alleen het belangrijke deel vermeld:

```
;;; Pronunciation of the numerical words:
;; !-------------------------------------------------------------
;; every word is accented to make sure the intonation module
;; tries to pronounce them explicitly.. or something... :)
(lex.add.entry '("nul" nn (((n ux l) 1)((#) 0))))
(lex.add.entry '("een" nn (((e: n ) 1)((#) 0))))
```

```
(lex.add.entry '("twee" nn (((t V[) 0) (( e: ) 1)((#) 0))))
(lex.add.entry '("drie" nn (((d R i: ) 1)((#) 0))))
(lex.add.entry '("vier" nn (((v i: R ) 1)((#) 0))))
(lex.add.entry '("vijf" nn (((v Ei f ) 1)((#) 0))))
(lex.add.entry '("zes" nn (((z e s ) 1)((#) 0))))
(lex.add.entry '("zeven" nn (((z e:) 1)(( v & n #) 0))))
(lex.add.entry '("acht" nn (((a x t ) 1)((#) 0))))
(lex.add.entry '("negen" nn (((n e:) 1)((G & n #) 0))))

(lex.add.entry '("zegen" nn (((z e:) 1)((G & n #) 0))))
(lex.add.entry '("even" nn (((e:) 1)((v & n #) 0))))
(lex.add.entry '("eveneen" nn (((e:) 1)((v & n) 0)((e: n) 1))))
```

B.5 festvox/net_dutch_frans_durdata.scm

Dit is het uiteindelijke duratie model, zoals gebruikt door `festvox/net_dutch_frans_duration.scm`. De genoemde tijden zijn na uitvoerig "tweaken" bepaald op basis van wat het beste klonk.

```
(set! net_dutch_frans::phone_durs
'(
   ;;; PHONE DATA
   ;; name zero mean in seconds e.g.
   (# 0.0 0.12)
   ; all other phones on net_dutch phoneset

   (ux 0.0 0.070) ;; nUl
   (e: 0.0 0.070) ;; EEn
   (a  0.0 0.060) ;; Acht
   (i: 0.0 0.100) ;; drIE
   (Ei 0.0 0.175) ;; vIJf
   (e  0.0 0.100) ;; zEs
   (&  0.0 0.030) ;; negEn

   (n  0.0 0.050) ;; Nul
   (V[ 0.0 0.060) ;; tWee
   (t  0.0 0.060) ;; Twee
   (d  0.0 0.025) ;; Drie
   (R  0.0 0.040) ;; dRie
   (l  0.0 0.070) ;; nuL
   (s  0.0 0.060) ;; zeS
   (x  0.0 0.100) ;; aCHt
   (G  0.0 0.060) ;; neGen
   (f  0.0 0.100) ;; vijF
```

```
   (v  0.0 0.090) ;; Vijf
   (z  0.0 0.060) ;; Zes

))
```

B.6 festvox/net_dutch_frans_tokenizer.scm

De standaard Festival met Nextens combinatie herschrijft cijfers als 21 naar "eenentwintig". Om dit uit te spreken kunnen fonemen nodig zijn die wij niet hebben opgenomen. Ons idee is regels te schrijven die "eenentwintig" uitspreken als "twee een". Dit is onelegant, maar garandeerd wel dat alle mogelijke cijfers uitgesproken kunnen worden.

Dit is de resulterende tokenizer om cijfers om te zetten in hun Nederlandse woorden. De tokenizer zal automatisch sequenties als "345" omzetten in "drie vier vijf". De code bevat geen Nextens onderdelen, maar is afgeleid van een voorbeeld uit de festvox documentatie.

```
(set! net_dutch::digit_names
    '((0 "nul")
      (1 "een")
      (2 "twee")
      (3 "drie")
      (4 "vier")
      (5 "vijf")
      (6 "zes")
      (7 "zeven")
      (8 "acht")
      (9 "negen")))

(define (net_dutch::number token name)
  "(net_dutch::number token name)
Return list of words that pronounce this number in dutch."
  (mapcar (lambda (d)
    (car (cdr (assoc_string d net_dutch::digit_names))))
      (symbolexplode name))
)
```

C Gebruikte documentatie

- *Building Synthetic Voices*
  http://festvox.org/bsv/

- *Festival Manual*
  http://festvox.org/docs/manual-1.4.3/festival_toc.html

- *Design of a Frisian Speech Synthesizer*
  http://www.fon.hum.uva.nl/IFA-publications/Others/FrisianTTS/spraaksynthesiser.pdf

D Wie deed wat

Stem	Frans Adriaans
Opnames	Marijn Koolen, Frans Adriaans
Segmentatie in wav files	Marijn Koolen, Frans Adriaans
Festival code	Ork de Rooij
"Tweaken" van het resultaat	Ork de Rooij
Documentatie hoofdstk. 1, 2 en A	Frans Adriaans, Marijn Koolen
Documentatie hoofdstk. 3, 4 en B	Ork de Rooij