Simon Dahlberg

Stavelseindelare för Europeisk Portugisiska

Abstract

Ett steg i ledet för TTS-teknik (Text-to-Speech), är att skapa ett program som kan dela in varje ord i stavelser. Detta program implementerar en helt ny algoritm för en sådan stavelseindelare för europeiskt portugisiska. Den är helt regelbaserad och utför sina analyser på grafemnivå, det vill säga den går igenom bokstav för bokstav (och hanterar alltså bara text, inte tal). Resultatet är en till synes väldigt korrekt stavelseindelning, med strax över 99,51% korrekthet.

Introduktion

Vad en stavelse är

En stavelse är, enkelt beskrivet, en del av ett ord, vars kärna, både i svenska och europeisk portugisiska (som nedan alltid kommer åsyftas bara som portugisiska) alltid består av ett vokalljud. Utöver kärnan kan en stavelse också vara försedd med en ansats - de konsonanter som föregår kärnan - och/eller en coda - de konsonanter som efterföljer kärnan. Det portugisiska ordet *por* (på svenska *för*) består alltså bara en av stavelse, där *p* är ansatsen, *o* utgör kärnan, och *r* står för codan.

Det som komplicerar stavelseindelningen är därför när ett ord består av flera (åtskilda) vokaler, och därmed flera stavelser. Om vi lägger till *ta* på det givna exempelordet, får vi *porta* (*dörr*). Det initiala *p:*et kan förstås bara utgöra första stavelsens ansats, men skall *rt* tillhöra sista stavelsens ansats eller den första stavelsens coda, eller ska skiljelinjen dras mitt emellan dem så att *r* blir coda och *t* ansats i olika stavelser?

Komplexiteten bakom portugisiskans regler för stavelseindelning

Innan detta programs algoritm kunde börja skissas fram, behövdes ett ställningstagande till vilka stavelseregler programmet skulle rätta sig efter. Det finns nämligen skilda åsikter bland lingvister om hur dessa bör se ut i portugisiskan. Hub Faria (175-180: 2007) hävdar att när det handlar om grupper om två konsonanter (såsom i *a**br**ir*), så skall dessa skiljas åt endast om den första bokstaven också skulle kunna avsluta ett ord. De exempel på bokstäver som ges är l, r och s, men också m, n och x skulle kunna inkluderas i den listan (även z kan avsluta ett portugisiskt ord, men inuti ett förekommer aldrig z följt av annan konsonant). I alla andra fall ska tvåkonsonantsgrupper tillhöra följande stavelses ansats. Tar vi en titt på priberam.pt (Portugisiskans största online-ordbok), ser vi dock att de delar in ordet *afta* i *af-ta* (https://www.priberam.pt/dlpo/afta), och inte som *a-fta*, som Hub Faria hävdar (bindestrecken står här alltså för gränsen mellan stavelserna).

Oliveira, C et al (2005) (http://pfonetica.web.ua.pt/files/publicacoes/3CFE_noprelo.pdf) beskriver, i sin rapport om en ny stavelseindelare, ytterligare ett sätt att dela in samma ord (och ord med liknande konsonantkluster), nämligen att låta *f* vara ansats till en stavelse där kärnan är en "tom" (eller neutral) vokal - *ø*. Denna tomma vokal anges ha sin grund i en fonologisk aspekt, nämligen att konsonantmötet inte är naturligt nog för att låta talare gå från den första konsonanten till den andra utan att något vokalljud kommer emellan. Ljudet må vara svagt, näst intill ohörbart, men de konstaterar ändå att en konsonantgrupp som *br* eller *fl* är naturligare och därför inte får någon tom vokal emellan sig.

Oliveira tar också upp komplexiteten med flera vokaler i följd. Som exempel kan vi ta ordet *cheio*, där *i* uttalas som en halvvokal, ungefär som *j* i svenskans *nej.* Detta *i* kan därför inte vara kärnan i en egen stavelse, utan får istället vara den mindre viktiga delen i antingen första eller andra stavelsens kärna. Frågan blir alltså om man ser *i*:et som del av en fallande (*ei*) eller stigande diftong (*io*).

Data

Då det handlar om ett regelbaserat program, vars algoritm utgår ifrån lingvistiken, så behövs ingen dataprocessering i förhand. Det finns alltså ingen träningsdata.

Programmets algoritm tar ett ord som indata, varpå utdatan blir samma ord stavelseindelat, med ytterligare information om vilka bokstäver som är ansats, vilka som är kärna och vilka som är coda. Se följande exempel:

Indata - *excelente*

Utdata:

N e
C x
O c
N e
O l
N e
C n
O t
N e
ex-ce-len-te

N står för Nucleus, kärna, medan O står för Onset, ansats, och C för Coda.

Programmet ger också användaren valet att köra igenom en hel fil textfil och avstämma resultatet mot en manuellt utförd stavelseindelning, för att ange träffsäkerheten (accuracy), det vill säga hur många av alla ord som blev rätt. Det är alltså inte vilken textfil som helst utan bara den som redan ligger i samma projekt (*plvrpt.txt*), med den manuella stavningsindelningen inlagd i ett specifikt format som programmet kan läsa. Denna textfil innehåller mestadels ord med komplex stavelseindelning. Ytterst få ord har en enda stavelse, och fördelningen av stavelsetyperna (O-N-C, O-N, N-C) är någorlunda jämnt

fördelade. I verkligheten är O-N överlägset vanligast, men för att testa programmet ändamålsenligt har svårare ord inkluderats. Orden i sig är hämtade delvis från en tidning i artikel i tidning Público och delvis från korpuset Corpus Português Fundamental (länkarna finns att se i Källor)

Metod

Klasser och attribut

Programmet är skrivet i Python. Det har två klasser: Ord och Stavelseindelare. Klassen Ord har tre attribut:

self.ordet, self.stavelsegraenser och self.stavelsekomponenter.

Det första tar själva strängen från indatan.

Det andra är en lista som innehåller siffror, vilka var och en anger sista bokstaven i en stavelse. Varje siffra motsvarar alltså ett index i strängen.

Det tredje är också en lista, men med strängar. Varje sträng är en bokstav, ‘O’, ‘N’ eller ‘C’, som alltså motsvarar stavelsekomponenten. Denna lista får samma längd som ordet självt, eftersom elementet på index x säger vad för stavelsekomponent bokstaven i ordet med samma index har.

Stavelseindelare har bara ett attribut: self.vokaler, vilket är en lista som innehåller alla vokalerna i portugisiskan (samt tom vokal), med och utan accent, tilde och cirkumflex. När någon av metoderna kontrollerar huruvida en viss bokstav är en vokal eller ej så gör den kontrollen mot denna lista.

Reglerna för Stavningsindelning

De regler som fastställdes för detta program landar i var man ska dra gränsen mellan två stavelser i en vokalgrupp och i en konsonantgrupp respektive. Reglerna definieras i sin tur efter gruppens storlek. En vokalgrupps storlek är alltså antalet vokaler som kommer i följd innan en konsonant hittas eller ordet tar slut, och vice versa för konsonantgrupper. Reglerna har sina fonologiska grunder, men analysen görs mot grafemet (den individuella bokstaven). Nedan beskrivs det med en sorts pseudokod, där varje indentering står för en indentering i Python. För tydlighetens skull har färger lagts till för att ange nivån på if-satsen. Röd är längst ut, blå är inuti en röd, och brun är inuti en blå.

För vokalgrupper gäller följande:

Om vokalgruppens storlek är 1 (alltså en enda vokal) - ingen gränsdragning.

Exempel: *porta -> por-ta*

Om storleken är 2:

Om det handlar om en diftong (alltså att den första eller den andra vokalen egentligen är en halvvokal) - ingen gränsdragning.

Exempel: *cheiro -> chei-ro*

Annars - dra gränsen mitt i.

Exempel: *tia -> ti-a*

Om storleken är 3:

Om det handlar om en triptong (alltså både första och tredje vokalen är en halvvokal), ingen gränsdragning.

Exempel: *contribuiu -> con-tri-buiu*

Om andra vokalen är en halvvokal, dra gränsen mellan den och nästa vokal.

Exempel: *raio -> rai-o*

Om bara första eller bara tredje vokalen är halvvokal, dra gränsen mellan de två som inte är halvvokaler.

Exempel: *poeiro -> po-ei-ro*

Om ingen av vokalerna är halvvokal, dra gränsen både mellan första och andra, och mellan andra och tredje.

Exempel: *baía -> ba-í-a*

Om storleken är 4, dra gränsen mellan tredje och fjärde vokalen.

Exempel: *uruguaio -> u-ru-guai-o*

Tilläggas skall att de enda två grafem som kan klassas som halvvokaler är *u* och *i*, med två undantag: *o* efter *ã*, och *e* efter *ã* eller *õ*. Huruvida ett visst *u* eller *i* inuti en vokalgrupp är halvvokal eller vokal, bestäms av ett antal if-satser som analyserar huruvida vokalen i fråga är betonad eller ej. Om den visar sig vara betonad, är den inte en halvvokal. Huruvida *o* och *e* kan klassas som halvvokaler i andra sammanhang än de angivna undantagen är föremål för diskussion.

För konsonantgrupper gäller följande:

Om konsonantgruppens storlek är 1 (alltså endast en konsonant) - dra gräns precis före den. Detta betyder alltså att denna konsonant blir den första bokstaven i sin stavelse.

Exempel: *caro -> ca-ro*

Om storleken är 2:

Om första konsonanten är *r*, *s* eller *x* - dra gränsen precis mitt i konsonantgruppen. Detta betyder alltså att r/s/x blir sista bokstaven i föregående stavelse, och den andra konsonanten i gruppen blir den första i nästa stavelse.

Exempel: *excursão -> ex-cur-são*

Om första konsonanten är *n* eller *l*:

Om andra bokstaven är *h*, dra gränsen före konsonantgruppen.

Exempel: *linha -> li-nha*

Annars dra den mitt i.
Exempel: *cunhado -> cu-nha-do*

Om första konsonanten är *m*:
Om andra är *p* eller *b*, dra gränsen mitt i.
Exempel: *com-pa-rar -> com-pa-rar*

Annars dra den före konsonantgruppen (exempel saknas).

Annars alltid före konsonantgruppen.
Exempel: *abrir -> a-brir*

Om storleken är 3:
Om andra bokstaven är s - dra gränsen efter detta s.
Exempel: *instalar -> ins-ta-lar*
Annars dra det mellan första och andra konsonanten.
Exempel: *concha -> con-cha*

Om storleken är 4 eller mer, dra gränsen mellan andra och tredje konsonanten.
Exempel: *obstruir -> obs-truir*

Dessutom är det så att om den andra bokstaven i en tvåbokstavig ansats inte är *l*, *r* eller *h*, skall dessa två konsonanter vara ansats i var sin stavelse. För den första blir kärnan en "tom" vokal, *ø*. Den andra konsonanten blir då ansats för nästa stavelse. Exempel: *psicologia -> pø-si-co-lo-gi-a*

Det bör också tilläggas att när *m* förekommer som sista bokstav i ett ord, så klassas det som halvvokal, vilket i praktiken betyder att det blir ytterligare en kärnkomponent i sin stavelse.

Anledningen till att just dessa regler fastställts är för att de förbättrar förutsättningarna för projektet och möjliggör en hög träffsäkerhetsgrad, samtidigt som resultatet håller sig väldigt nära vad många lingvister skulle säga vara korrekt stavningsindelning. Träffsäkerheten kommer tas upp i avsnittet om resultat och utvärdering.

Algoritmen

Algoritmen har skrivits utifrån de regler för stavningsindelning som fastställts för detta program. De landar alltså i var man ska dra gränsen mellan två stavelser, det vill säga vilken siffra man ska lägga till i listan self.stavelsegraenser för Ord-objektet i fråga.

Klassen Stavelseindelare har flera metoder som samarbetar för att komma fram till vilka siffror man ska skicka till den.

Efter den processen, går en annan metod igenom stavelse för stavelse för att definiera komponenterna i dem. Då fylls listan self.stavelsekomponenter. Vokalerna i varje stavelse får N (kärna), konsonanterna dessförinnan får O (ansats), och konsonanterna efter kärnan får C (coda), med ovan förklarat undantag för ordfinalt *m.*

Därefter går ytterligare en metod igenom den listan, och undersöker huruvida någon stavelseansats innehåller ett konsonantpar med onaturlig övergång. Det betyder i praktiken att den andra bokstaven inte är *l*, *r* eller *h*. Som förklarat ovan, så blir den första konsonanten däri ansats till en egen stavelse, vari kärnan är en tom vokal. Det betyder att bokstaven *ø* läggs till på rätt plats i ordet, och därtill ytterligare en stavelsegräns och stavelsekomponent på rätt plats i respektive lista.

Resultat och utvärdering

Av de 205 ord som finns i textfilen med (av underskriven) manuellt utförd stavelseindelning, så blev 204 rätt (det vill säga likadana som den manuellt gjorda annoteringen). Det ger en träffsäkerhet (accuracy) på 99,51%. Det skulle förstås vara ännu högre om textfilen inte var skriven just för att testa programmet, framför allt med tanke på att den allra vanligaste stavelsetypen är O-N (Oliveira C, 1-2: 2005), vilken är väldigt enkel då man då inte ens tillämpar algoritmerna för att behandla vokal- och konsonantgrupper som är två bokstäver lång eller mer. Oliveira beskriver ett program som överstiger 99,5%, men med en helt annan algoritm än denna. Orsaken till felen där uppges till stor del vara komplexa vokalgrupper. Det programmet kördes dock mot ett mycket större korpus, varför det är svårt att rättvist jämföra träffsäkerheten, men vi kan ändå konstatera att problemen är olika. I vilket fall måste resultatet vara högt nog för att ses som tillräckligt för att förtjäna att prövas i ett större TTS-program.

Det enda ord där vårt program felade var *exactamente.* Problemet här är *ct*, som innan slutfasen står som ansats med ett onaturligt konsonantpar, och därför tilläggs den tomma vokalen, med tillkommande stavelsegräns och stavelsekomponent. Saken är att *c* i *exactamente* i verkligheten inte uttalas, varför det egentligen inte är tal om någon extra tom vokal. Därför annoterades ordet manuellt till att ha tre stavelser, inte fyra som programmet tar fram. Om programmet kördes mot ett större korpus, skulle detta problem uppstå i alla ord med sådana stumma konsonanter. Samtidigt som detta problem enligt portugisiskans nya stavningsreform inte ska finnas kvar, då de stumma *c*:na och *p*:na har tagits bort, så är det självklart bättre att kunna behandla ord som är skrivna enligt de “gamla” stavningsreglerna.

Slutsats

Överlag är programmet väldigt kompetent vad gäller det som den ämnades kunna utföra, alltså att dela in stavelser i portugisiska ord enligt de för de fastställda reglerna.

Vill man emellertid ha ett program som även behandlar ord med stumma *c*:n och *p*:n så behövs en lista över de specifika ord som innehåller det. Programmet måste då reagera på dessa ord för att inte felaktigt skjuta in en tom vokal, vilket är fullt genomförbart. Det skulle snarare bli ytterligare en regel, med en textfil på fullt överblickbar storlek.

I vilket fall, måste resultatet på 99,51% ses som väldigt lyckat.

Källor

Hub Faria, Isabel et al. (2007). *Introdução à Linguística Geral e Portuguesa*. 2nd ed. Lissabon: Caminho. 170-180.

Artikel från Público (1 december 20:00)
(https://www.publico.pt/2016/12/01/economia/noticia/ppp-rodoviarias-da-madeira-custaram-490-milhoes-de-euros-em-tres-anos-1753360)

Corpus Fundamental Português (6 december 12:30)
http://www.clul.ul.pt/pt/recursos/84-spoken-corpus-qportugues-fundamental-pfq-r

Artikel av Catarina Oliveira et al (2005):
http://pfonetica.web.ua.pt/files/publicacoes/3CFE_noprelo.pdf