Manusverktyg

STTS Södermalms talteknologiservice
Folkungagatan 122 2tr, 116 30 Stockholm
http://stts.se

13 april 2021

Innehåll

1 Inledning **2**

2 Urvalsprocessen **2**

2.1 Normalisering 2

2.1.1 Parsning av Wikipediaartiklar 3

2.2 Filtrering 3

2.2.1 Domäner 4

2.2.2 Tillgängliga filterkriterier 5

2.3 Urval 6

2.3.1 Mått på tillförd fonetisk variation 6

2.4 Tillgängliga särdrag för urval 7

3 Att använda manusverktyget **7**

3.1 Inläsning av normaliserad korpus 7

3.2 Filter och urval 8

3.3 Outputformat vid urval 8

1 Inledning

För att skapa ett bra manus för inspelning av talsyntes (text-till-tal) eller taligenkänning (tal-till-text), vill man ta fram en fonetiskt balanserad text, där så många bra kombinationer av ljud som möjligt finns med. Man vill också att texten skall vara möjlig att läsa högt utan för stora problem. Det kan också finnas önskemål om text från speciella typer av text, eller text som innehåller vissa typer av ord.

Manuskriptverktyget tar en textkorpus, och väljer ut en delmängd av valfri storlek med önskade egenskaper.

Det som gör uppgiften komplicerad, är att texten som väljs består av bitar (meningar) som inte är oberoende, eftersom manuset utgörs av kombinationen av alla utvalda meningar. Om en mening skall väljas ur korpusen eller ej kan bero på vilka meningar som tidigare har valts. Detta gör det till en potentiellt mycket krävande uppgift om man förväntar sig en fullständig sökning (alla kombinationer av möjliga delmängder måste jämföras mot varandra, enligt något mått). Därför är det rimligt att dela upp uppgiften i olika delar, och ta fram olika bitar av manuset i taget.

Ett annat problem, är att man egentligen är ute efter hur texten låter när den läses upp, och det är inte troligt att man har ett uttalslexikon som täcker alla ord i en stor korpus. Därför kan det vara en rimlig kompromiss att anta att stavningen av ord någorlunda avspeglar uttalet, och att man försöker få med så många olika ord som möjligt, och därmed indirekt få en fonetisk spridning. Det finns även andra sätt att öka den fonetiska variationen utifrån stavningen, exempelvis genom att maximera antalet olika övergångar mellan ord (tex tre tecken åt varje håll, etc). Detta fungerar olika bra för olika språk.

Man kan också ha ett moment där man ser till att korrekt transkribera de utvalda meningarna, och iterativt leta i den kvarvarande korpusen efter särdrag som saknas. Detta ingår inte i nuvarande manusverktyg, men om man har fonetiska transkriptioner, går det att lägga till dem som särdrag i korpusen.

2 Urvalsprocessen

Det aktuella manusverktyget behandlar korpusen i tre steg: normalisering, filtrering och urval.

2.1 Normalisering

Oavsett om textkorpusen är uppmärkt eller strukturerad på något vis eller ej, börjar man med att normalisera den till ett format man vill ha, exempelvis genom att dela upp den i meningar. Detta kan vara mer eller mindre problematiskt, beroende på varifrån korpusen kommer och i vilket skick den är. En äldre korpus kan exempelvis ha någon gammal teckenkodning som måste konverteras till UTF-8, etc.

2.1.1 Parsning av Wikipediaartiklar

För det här projektet har vi valt att använda Wikipediaartiklar som källdata. Att konvertera Wikipediaartiklar till råtext är inte trivialt, på grund av den uppmärkning som finns i texten.

Vi har valt att använda ett befintligt program, WikiExtractor.py[1], som vi har modifierat något. Detta är inte utan problem, men det är det bästa verktyg vi hittat. Stor möda har lagts ner av skaparna av WikiExtractor.py, men det räcker ändå inte för att på ett generellt sätt kunna extrahera den faktiska artikeltexten ur en Wikipediadumpfil. Det finns för stor variation i uppmärkningen av texter. Dessutom går parsningen ganska långsamt.

Ett svårhanterat problem är expansionen av mallar, som används flitigt för att exempelvis generera årtal, datum och liknande. Användningen av mallar är så varierad att det är i det närmaste hopplöst att skriva ett program som kan hantera alla mallar i en dumpfil korrekt.

Vi har tills vidare valt att inte expandera mallar, utan att helt sonika filtrera bort meningar som innehåller oexpanderade mallar. Det är otillfredsställande eftersom det innebär att all text i artiklarna inte kommer med i manusskapandet. Att lösa detta på ett bättre sätt är framtida arbete.

Det indataformat vi har valt för lexikonverktyget är det som produceras av WikiExtractor.py. Det betyder att om man har en annan korpus än Wikipediadumpar, får man konvertera den till det formatet. (Alternativt kan man ganska lätt anpassa inläsningen till andra format om så skulle behövas.)

2.2 Filtrering

Man vill så gott som alltid filtrera textkorpusen, för att få bort sånt som inte är önskvärt. Det kan handla om rester av uppmärkning (exempelvis HTML), konstiga/ovanliga tecken som inte ingår i språkets alfabet, för långa meningar, för korta meningar, meningar med för många skiljetecken, meningar med för få skiljetecken, etc. (Notera att ”kort” respektive ”lång” mening är väldigt grova mått, och riskerar att kasta bort bra material.) Detta är helt beroende på hur textkorpusen ser ut, och vilka specifikationer man har för det slutliga manuset.

När man har en normaliserad korpus med oönskade delar borttagna, kan man använda fler filter för att hitta egenskaper som är önskvärda.

Detta kan exempelvis vara att få med ord från specifika domäner, som veckodagar, månader, viktiga städer, eller vanliga namn. Korpusen kan delas in i delmängder med önskade egenskaper, och ur dessa kan de mest lovande meningarna väljas tills man har täckning för det man vill ha.

Man kan tillämpa olika filter i olika iterationer. Rent allmänt är det klokt att ta fram manuset i mindre delar, som tillsammans sedan bildar ett komplett manus. Man får räkna med att manuset inte får en ”optimal” balansering av särdrag, men det är som tidigare antytts i praktiken en omöjlig uppgift.

Filtreringen i verktyget sker i två steg. I steg ett filtreras korpusen vid inläsningen. Det betyder att vissa texter eller meningar aldrig läses in i databasen, utan väljs bort direkt.

[1] `http://medialab.di.unipi.it/wiki/Wikipedia_Extractor`

Det kan handla om att ta bort text som innehåller oönskad uppmärkning, eller enheter som innehåller för många ord (eller ”tokens”). Denna filtrering är hårdkodad, och tänkt att vara relativt generell, men går att modifiera vid behov.

En annan typ av filtrering sker efter att korpusen sparats i databasen, och utförs med hjälp av filterkriterier som användaren väljer. Det kan handla om meningslängd, minsta tillåtna ordfrekvens per mening, antal skiljetecken, ord med speciella egenskaper, etc.

Varje filtrerad del av databasen kallas en *batch*.

Ytterligare ett mått kan vara läsbarhet: man vill att manuset skall vara så lätt som möjligt att läsa högt. Det är inte helt uppenbart hur man automatiskt avgör hur en mening är lättare att läsa än en annan, men man kan använda saker som meningslängd, ordlängd och ordfrekvenser. Det finns även ett mått för att beräkna läsbarhetsindex (LIX) för en text: `https://sv.wikipedia.org/wiki/L%C3%A4sbarhetsindex`, men det är inget vi har implementerat i verktyget.

Man kan också ta fram filter som slår till på udda kombinationer av bokstäver och bokstavskombinationer, som inte är typiska för språket. På så vis kan man identifiera potentiellt svåruttalade meningar.

2.2.1 Domäner

Om man är intresserad av att få med olika textdomäner, kan ordlistor användas som ett (trubbigt) verktyg. För vissa språk kan det vara lämpligt att använda Wikidata för att få ut kategorier av ord som kan antas ha med någon domän att göra. Exempelvis kan man söka efter namn på skådespelare, utifrån en lista på de skådespelare som har högst rankning i Wikidata. På samma vis kan man skapa en lista över städerna med högst rankning, för att få med vanliga namn på städer, etc. (Om man inte har tillgång till alla böjningsformer av de ord man vill ha, kan man leta efter de eftersökta orden också som prefix, och med lite tur få med mer av det man vill ha.)

Till manusverktyget hör exempellistor för några olika domäner, i katalogen `feat_data`: svenska namn på sporter, väderord, städer, vanliga förnamn, vanliga efternamn, med mera.

I nuvarande release finns följande domäner inlagda:

Namn	**Beskrivning**
bigram_top800	De 800 vanligaste bigrammen
int_place	Internationella platsnamn (på svenska och originalspråk)
se_calendar	Kalenderord för svenska (månader, dagar, med mera)
se_fem_name	Kvinnliga förnamn i Sverige
se_fem_name_top100	De 100 vanligaste kvinnliga förnamnen i Sverige
se_male_name	Manliga förnamn i Sverige
se_male_name_top100	De 100 vanligaste manliga förnamnen i Sverige
se_place	Platsnamn i Sverige
se_sports	Svenska namn på sporter
se_surname	Efternamn i Sverige
se_weather	Väderrelaterade ord

2.2.2 Tillgängliga filterkriterier

Nedan finns en beskrivning av de filterkriterier som finns att välja på i nuvarande release av manusverktyget. Informationen är på engelska, eftersom den är genererad direkt från programdokumentationen.

comma_count
Number of commas in a sentence
Args Two integers defining a legal interval
Example `0, 2`

digit_count
Number of digit expressions in a sentence
Args Two integers defining a legal interval
Example `0, 0`

exclude_batches
Exclude sentences from certain batches
Args List of batch names
Example `test_batch_1`

exclude_chunk_re
Illegal sentence pattern
Args Regular expression
Example `[\p{Greek}]`

lowest_word_freq
Lowest word frequency allowed in a sentence
Args One integer defining the frequency
Example `3`

paragraph_count
Choose sentences from texts (sources) containing a certain number of paragraphs
Args Two integers defining a legal interval
Example `10, 20`

sentence_count
Choose sentences from texts (sources) containing a certain number of sentences
Args Two integers defining a legal interval
Example `9, -1`

source_re
Required pattern for text source
Args Regular expression
Example `00$`

word_count
Number of words in a sentence
Args Two integers defining a legal interval
Example `4, 25`

chunkfeat_cats
Choose sentences included in pre-defined feature categories (typically domains)
Args List of feature categories selected from:

- bigram_top800
- int_place
- se_calendar
- se_fem_name
- se_fem_name_top100
- se_male_name
- se_male_name_top100
- se_place
- se_sports
- se_surname
- se_weather

Example `se_place`

2.3 Urval

Baserat på en eller flera filtrerade *batchar*, görs ett eller flera urval, som ingår i det slutgiltiga manuset. Tillvägagångssättet beskrivs nedan. Varje urval (eller delmanus) som skapats, sparas i databasen som ett *script*.

2.3.1 Mått på tillförd fonetisk variation

För att veta vilken av två meningar som ger mest ny information, behövs mått för detta.

Man behöver ett eller flera mått, som anger hur mycket ny fonetisk variation en mening tillför en existerande mängd meningar. Ett dåligt valt mått kan få som konsekvens att man alltid väljer den längsta eller kortaste meningen, eller en med många felstavningar i.

Genom att skapa frekvenslistor på teckensekvenser i korpusen (n-gram) kan man se till att så många vanliga sekvenser som möjligt finns med i manuset. Exempelvis kan man vilja ha så många olika övergångar mellan ord som möjligt, och så många olika finala sekvenser innan paus (efter kommatecken eller i slutet av mening) som möjligt. Det betyder att en mening som har en viss teckensekvens som inte redan finns i tidigare valda manusmeningar är mer värdefull än en som bara innehåller sekvenser som redan finns i redan valda meningar.

När man har med frekvenslistor över fritext att göra, är det klokt att vara misstänksam mot särdrag med låg frekvens, eftersom det ofta kan röra sig om rena fel i texten, eller helt enkelt väldigt ovanliga fenomen. (Det är dock inte helt oproblematisk att ta bort alla lågfrekventa särdrag, eftersom ovanliga fall i den aktuella korpusen kan visa sig viktiga i annan text som

man ännu inte har sett.)

Man vill ha med så många vanliga saker som möjligt, men inte *samma* vanliga saker. Exempelvis vill man ha med så många *olika* vanliga ord som möjligt. Man vill också ha med ovanliga förekomster, men inte för många ovanliga, för det tenderar att välja ut konstiga meningar som är svåra att läsa.

Ju fler olika saker man tar med i måttet, desto svårare är det att förstå vad det egentligen betyder. Man skall alltså vara försiktig med dessa mått. Det kan vara en idé att använda flera olika mått, och köra urvalet i olika omgångar för varje mått.

Exempel på mått:

- antal nya ord i en mening
- antal önskade nya ord (enligt någon lista)
- unika nya kombinationer (exempelvis av suffix-prefix av ordpar)
- antal nya kombinationer av fonem (om man har tillgång till fonetisk representation av texten)

Man kan också använda sig av en slumpkomponent, som grupperar meningarna i slumpmässiga grupper, och använder dessa grupper som minsta enhet, för att inte luras av mått som gör urvalet för snävt.

2.4 Tillgängliga särdrag för urval

Namn	**Beskrivning**
bigram	Tvåbokstavskombinationer
trigram	Trebokstavskombinationer
bigram_top800	De 800 vanligaste bigrammen
bigram_transition	Bigram i övergångar mellan ord
initial_bigram	Meningsinitiala bigram
final_trigram	Meningsfinala trigram
words	Ord

3 Att använda manusverktyget

Information om hur man använder manusverktyget finns i projektets README-fil. Nedan följer en kortfattad beskrivning av de olika stegen.

3.1 Inläsning av normaliserad korpus

Först behöver man läsa in en normaliserad textkorpus. För att läsa in en textkorpus behöver man en eller flera filer med artiklar. Varje artikel ska ligga i ett `doc`-element. Elementets `url`-attribut sparas som artikelns *källa*. Inuti elementet ligger texten stycke för stycke, där stycken avgränsas med två radbrytningar.

Exempel:

```
<doc id="72324" url="https://sv.wikipedia.org/wiki?curid=72324" title="Almunge">
Almunge

Almunge är en tätort i östra delen av Uppsala kommun vid länsväg 282. En kilometer
öster om tätorten ligger Almunge kyrkby där Almunge kyrka, Almunge skola och en
bensinstation med livsmedelsaffär finns.

Orten Almunge hade ursprungligen namnet Lövsta men bytte namn när Upsala-Lenna
Jernväg invigdes 1884 med en station här då Lövsta ansågs för vanligt varför
grannbyns ovanligare namn användes istället.

I Almunge finns livsmedelsaffärer, värdshus, skola, vårdcentral, idrott/simhall,
kyrka och bostäder. Från Almunge station kan man på sommaren ta Lennakatten som
går mellan Faringe station och Uppsala Östra.

</doc>
```

Mer information finns i README-filen under avsnittet *Set up DB*.

3.2 Filter och urval

Manusverktyget är paketerat i en CLI med namnet `scripttool`. Man kan anropa verktyget på olika sätt, antingen för att tillämpa filterkriterier, eller för att göra ett manusurval utifrån tidigare filtrerade delar, eller köra filtrering och urval direkt i samma körning.

Man kan också filtrera i flera steg, det är förmodligen ett ganska vanligt scenario.

3.3 Outputformat vid urval

- Inställningar för filter och urval
- Lista på meningar inkl källhänvisning (för Wikipedia tex artikelid eller länk till artikeln)
- Enkel metadata om urvalet (antal meningar i input batch/er, antal meningar i outputen)

Mer information finns i README-filen under avsnittet *Scripttool*, samt i hjälp-funktionen för kommandot `scripttool`.