|  |  |
| --- | --- |
| Arbejdspakke, nummer og navn | WP2.2 |
| Aflevering, navn | Tekniske metoder og annotationer revideret |
| Afleveringsfrist | 30. april 2010 |
| Afleveringsdato | 7. maj 2010 |
| Projektleder for arbejdspakken | Sussi Olsen |
| Deltagere | Jakob Halskov, Dorte Haltrup Hansen, Anna Braasch |

Tekniske metoder og annotationer for

det fagsproglige korpus i DK-CLARIN, arbejdspakke 2.2

Revideret pr. 7. maj 2010

Indhold

[1. Tekniske løsninger: Korpusindsamling 1](#_Toc260997028)

[1.0 Om indsamlingen generelt 1](#_Toc260997029)

[1.1 Indsamling af elektroniske dokumenter. 1](#_Toc260997030)

[1.1.1 Downloads af internetsider 1](#_Toc260997031)

[1.1.2 Andre elektroniske dokumenter 3](#_Toc260997032)

[1.2 Kvalitetskontrol af tekstformat 3](#_Toc260997033)

[1.3 Rengøring af dokumenterne 5](#_Toc260997034)

[1.3.1 Kvalitetskontrol af dokumentrengøring 6](#_Toc260997035)

[2. Tekniske løsninger: Korpusopmærkning 6](#_Toc260997036)

[2.1 Dokumentannotation (annotation med metadata) 6](#_Toc260997037)

[2.2 Tekstannotation (lingvistisk annotation) 8](#_Toc260997038)

[2.2.1 PoS-tagging 8](#_Toc260997039)

[2.2.2 Lemmatisering 8](#_Toc260997040)

[2.2.3 Term-tagging 9](#_Toc260997041)

[2.2.4 Kvalitetskontrol af tekstannotation 10](#_Toc260997042)

# 1. Tekniske løsninger: Korpusindsamling

## 1.0 Om indsamlingen generelt

Baseret på erfaringerne fra korpusindsamling i STO-projektet vil indsamlingen for de forskellige domæner og fag så vidt muligt ske parallelt. Dette skal dels sikre at tekstmaterialet er af forholdsvis synkron oprindelse på tværs af det samlede korpus, og samtidig afdække eventuelle domænespecifikke problemstillinger så tidligt som muligt i projektforløbet.

## 1.1 Indsamling af elektroniske dokumenter.

Vi vil koncentrere os om elektroniske tekster da det tekniske arbejde med papirdokumenter, såsom indskanning, OCR-genkendelse og efterfølgende korrekturlæsning vil blive for tidskrævende i forhold til udbyttet.

### 1.1.1 Downloads af internetsider

Helt generelt omfatter dette punkt tre faser, nemlig 1) download (spejling) af hjemmesiden, 2) bortfiltrering af uønskede tekstformater og 3) konvertering af ønskede tekstformater til et fælles format.

Her følger en tabel over ønskede og uønskede tekstformater, mulige problemstillinger samt relevante konverteringsværktøjer. Tabellen er ikke udtømmende, men udbygges løbende som projektet skrider frem.

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel 1: Tekstformater på internettet og relevante konverteringsværktøjer | | | |
| ***Format*** | ***Ønskværdighed*** | ***Særlige problemer*** | ***Konverteringsværktøjer*** |
| HTML | Ja | hvis der arbejdes med frames kan man ikke få vist kildekoden for den pågældende side (se fx [*http://www.astronomibladet.dk/*](http://www.astronomibladet.dk/)) | cpan.org: HTML::FormatRTF |
| XHTML | Ja | XHTML er lettere at tilgå end HTML, da det følger en mere stringent struktur (fx skal alle tags lukkes) | se HTML |
| PPT | Nej | Bør ikke indgå i korpusset, da det næppe kan betragtes som sammenhængende tekst (der er sjældent tale om det komplette informationsindhold). | - |
| WORD | Ja |  | fx: Abiword + script[[1]](#footnote-1) |
| PDF | Ja | Et populært og vigtigt format. Dog kan det være vanskeligt at bevare den oprindelige formatering ved konvertering til andre formater. | * Acrobat (CST har licens) * Abbyy Finereader 8.0 (DSN har livcens) * unix: XPDF (til TXT) * cpan.org: PDF::API2 * www.exegenix.com (til RTF, men kommercielt[[2]](#footnote-2)) |
| ASP, PHP, ... | Ja | Mere og mere indhold på internettet genereres dynamisk via databaseopslag på serversiden. Download af denne type tekst kan imidlertid være vanskelig da dokumenterne er dynamiske hvilket vil sige at de først genereres når man besøger dem. Download skal derfor foregå manuelt. | Se HTML |

En anden udfordring er håndteringen af det multimodale indhold på hjemmesider, nemlig billeder, lyd, video og så videre. Der er følgende niveauer i håndteringen af denne type indhold:

1. Multimodalt indhold downloades og bevares i den oprindelige form i søgegrænsefladen
2. Multimodalt indhold downloades, men vises ikke i den oprindelige form i søgegrænsefladen (kun i form af links/henvisninger)
3. Multimodalt indhold downloades ikke, men der oprettes henvisninger de steder i teksten hvor indholdet var placeret
4. Multimodalt indhold downloades ikke og der oprettes ingen henvisninger i teksterne

Selvom det kan medføre et tab af information og i værste fald gøre tekstmaterialet svært forståeligt vurderer arbejdspakkegruppen at arbejdet med at håndtere multimodalt indhold på niveau 1 og 2 ikke modsvarer gevinsten. Sådant indhold vil derfor blive håndteret på niveau 3 eller 4. Desuden beskæftiger wp 3.1. sig med multimodalt indhold.

**Tekniske løsninger**

Vi skal bruge værktøjer

1. til at registrere de forskellige tekstleverandører og lagre aktuelle kontaktinformationer samt statusinformationer om leverancer og modtagne aftaledokumenter
2. downloade/spejle hjemmesiderne
3. frasortere uønskede tekstformater
4. opbevare de originale tekster i en database
5. konvertere teksterne til RTF
6. konvertere RFT til det fælles standardtekstformat for WP2 (se http://korpus.dsl.dk/clarin/corpus-doc/text-format.pdf)
7. annotere teksterne ved at tilføje forskellige "spans" (primært tokenisering, lemmatisering, pos-tagging og term-tagging)
8. opbevare de annoterede tekster samt headere i en XML-database (se afsnit 4.1 om tekstheadere)
   1. En leverandørbase er allerede implementeret og taget i anvendelse på https://clarinja.javaprovider.net:7777/thirdparty/phpMyAdmin (user: clarinja, pass: clarin)
   2. Download/spejling kan ske med unix-værktøjet, WGET, der rekursivt kan skabe et lokalt spejlbillede af hjemmesider blot man giver det en URL og et antal parametre.
   3. Frasortering kan løses vha. simple scripts.
   4. En database til upload og opbevaring af de oprindelige tekster er udviklet og ligger midlertidigt på https://192.38.108.131/phpMyAdmin/
   5. Konvertering til det fælles standardtekstformat kræver udvikling af en ordhakker. En sådan er udviklet.
   6. Yderligere annotering af teksterne bør ske ved at tilpasse eksisterende værktøjer så de kan tage standardtekstformat (XML) som input og levere annotationslag (XML spans) som output. Dette arbejde er i fuld gang.
   7. Den endelige opbevaring af annoterede tekster (og tilhørende headere) vil, i WP 2.1 og WP 2.2's tilfælde, ske i XML-databasesystemet eXist, der allerede er tilpasset til formålet og installeret på DSL (inklusiv en redigeringsgrænseflade) samt i DK-CLARINs repositorie.

### 1.1.2 Andre elektroniske dokumenter

Problemstillingerne under punkt 3.1.2 modsvarer sandsynligvis dem under punkt 3.1.1. Dog vil der antageligvis være mindre støj, idet man ikke vil skulle håndtere særlige træk ved hjemmesider som fx menuer, rammer, links, javascripts, dynamisk genereret indhold osv. Dog vil man stadig skulle tage stilling til håndteringen af fx billedindhold.

**Tekniske løsninger**

Indsamlingen kan foregå enten manuelt eller med fx WGET (jf. punkt 3.1.1). Hvis der er tale om et begrænset antal dokumenter fra et stort antal hjemmesider med ret forskellig struktur, er det spørgsmålet om det kan svare sig at programmere en automatisk nedlastningsløsning. Måske vil det ofte være tilfældet at dokumenterne forefindes i en mere tilgængelig form som det vil kræve en personlig henvendelse at få adgang til. Og under alle omstændigheder skal tekstleverandørerne alligevel kontaktes.

## 1.2 Kvalitetskontrol af tekstformat

Et grundlæggende spørgsmål er hvordan man tjekker at konverteringen til XML ikke har ødelagt teksterne. Og hvis en tekst er gået i stykker, hvad gør man så ved den? Et andet vigtigt spørgsmål er hvordan man sikrer sig mod "forurening" af korpusset via eksempelvis engelsksprogede tekster. Endelig bør der etableres en kontrol som sikrer at de tekster der indlemmes i korpusset, rent faktisk er fagsproglige og ikke almensproglige.

Arbejdspakkegruppen har forsøgsvis implementeret de følgende automatiske kvalitetstjek:

Tjekkene inddeles i to grupper, de der måler læsbarhed og graden af formelt sprog, og de der måler fagsproglighedsgraden i teksterne

**Læsbarhed/formelt sprog**

Ordlængde: % af ord over 8 bogstaver)

Sætningslængde: gennemsnitlig sætningslængde

Passive verber: % af antal s-passiver

Personlige pronominer: % af personlige subjektspronominer

Top 1000 A, V, N: % af de 1000 hyppigste A, V, N i almensproglig tekst

Top 1000 alle tokens: % af de 1000 hyppigste tokens i almensproglig tekst

**Fagsproglighed/videnstæt**

Dokumentets termrighed: antal ukendte types/alle types

Tæthed af eksplicit viden: vidensmønstre / 1000 tokens)

6) Brug af engelsk: ukendte types i BNC / ukendte types

Definitionen på en "ukendt" type er en ordform som ikke forekommer i en databasetabel over et stort antal kendte ordformer (pt. ca. 2 mio. ordformer). De kendte ordformer stammer fra følgende kilder/referenceværker:

1) Opslagsord fra Retskrivningsordbogen 2001 (foldet ud i alle bøjningsformer)

2) Opslagsord fra DDO (minus pkt. 1)

3) Opslagsord i Dansk Sprognævns Ordsamling (minus pkt. 1-2)

4) Ordformer fra Korpus 90 (minus pkt. 1-3)

5) Ordformer fra Korpus 2000 (minus pkt. 1-4)

6) Ordformer fra Dansk Sprognævns Infomedia-monitorkorpus (minus pkt. 1-5)

Definitionen på et vidensmønster er en tekststreng som realiserer semantiske eller konceptuelle relationer (foreløbigt kun synonymi og hyponymi/hyperonymi) mellem begreber i tekst. Foreløbigt anvendes de følgende 16 vidensmønstre, men denne liste er langt fra udtømmende:

Synonymi: også kendt som / også kaldet / kaldes (også) / det vil sige / dvs. / refererer til

Hyponymi: så som / for eksempel / i form af / omfatter / fx

Hyperonymi: er en slags / er en type / og andre / defineres som / er defineret som

Alle målene kan anvendes som indikatorer på fagsproglighedsgraden af et dokument. Eksempelvis antages det at ekspert-ekspert-tekster scorer markant lavere end almensproglige tekster i forhold til mål nummer 1, men markant højere i forhold til mål nummer 2-6. For tekster forfattet af eksperter/semieksperter og rettet mod semieksperter/lægmænd gælder samme antagelse omend værdierne sandsynligvis vil ligge mellem de meget specialiserede dokumenter og de almensproglige dokumenter (med undtagelse af tætheden af eksplicit viden som antages at være højest i de mere formidlende fagsproglige tekster). Derimod er tekster som fx har en alt for lav lix eller en alt for lav termrighed, med stor sandsynlighed for almensproglige til at blive indlemmet i korpusset.

Med hensyn til de (mere eller mindre) engelsksprogede tekster, så vil mål nummer 6 kunne identificere disse dokumenter, hvilket allerede er afprøvet i praksis (BNC står for British National Corpus).

Hvad angår den ortografiske integritet i dokumenter som er blevet scannet eller som er blevet automatisk konverteret fra deres oprindelige tekstformat, så vil problemer på dette punkt kunne spottes ved hjælp af mål nummer 2 og mål nummer 4. Konverteringsfejlene vil nemlig enten resultere i ekstremt mange ukendte types og/eller ekstremt lange ord kombineret med uforholdsmæssigt lange sætninger.

Foreløbige prøvekørsler af denne metode på tekster fra medicin/sundhed og klima/miljø domænerne har givet gode resultater. Precision er 100 %, kun gode fagsprogstekster gik igennem, mens recall er på 55, nogle tekster der viste sig at være udmærkede fagsproglige tekster, blev skilt ud. Der forestår nu at afprøve metoden på andre domæner og evt. justere grænseværdierne. Ligeledes kan det evt. blive aktuelt at indføre domænespecifikke kvalitetstjek/grænseværdier.

## 1.3 Rengøring af dokumenterne

I de dokumenter man har udvalgt til korpusset, skal meningsfuld tekst sies fra opmarkeringstekst. En stor del af denne opgave håndteres automatisk af konverteringsværktøjerne i tabel 1, men man bør have en klar definition af hvad der er meningsfuld tekst, og hvordan det repræsenteres. Hvordan repræsenterer man fx tekst fra en tabel? Hvad kommer først rækker eller kolonner? Og har overskrifter anden status end anden tekst?

Disse spørgsmål har arbejdspakkegruppen endnu ikke nogen endelige svar på. Som beskrevet i afsnittet om domæneselektion vil vi forsøge at undgå domæner med en høj tæthed af kunstsprog (fx formler, reaktionsligninger og lignende). Alligevel vil megen meningsfuld tekst uvægerligt optræde i eksempelvis tabelform. Det kan vise sig vanskeligt at finde konverteringsteknikker som bevarer tabelstrukturen, og dermed vil rækker sandsynligvis komme før kolonner.

Overskrifter bør have en anden status end resten af teksten. Det er nemlig vigtigt at undgå at deres tekstindhold bliver en del af den efterfølgende sætning idet tekstens ortografiske integritet dermed brydes, og der genereres uidiomatiske tekstsegmenter i korpusset.

I sidste ende skal teksterne overholde det fælles standardtekstformat som igen skal overholde TEI P5-standarden. Der er således ikke mulighed for at indføre arbitrære elementer og anvende dem til at opmærke eksempelvis tabeller, rækker og celler. Løsningen bliver således sandsynligvis at anvende det tilladte <p>-element med tilføjelse af forskellige attributter. Teksten bør nu være klar til lingvistisk annotation.

**Tekniske løsninger**

Kvalitetsmålene beskrevet under punkt 3.3 fordrer alle at tekstmaterialet er ordopdelt (tokeniseret) og indeholder sætningsgrænser. Disse to trin indgår i rengøringsfasen hvor CST's værktøjskasse generelt kan byde på mange eksisterende løsninger (se nedenfor). Fælles for alle eksisterende værktøjer er dog at de skal tilpasses til at tage standardtekstformat som input og generere spanlag som output. Dette er ikke nogen helt triviel udviklingsopgave.

Dog må man også forvente at der i et vist omfang skal laves nogle ad hoc-løsninger til specielle fænomener alt efter dokumenttype og dokumentkarakteristika. De enkelte løsninger implementeres i scripts der så vidt muligt kan genanvendes.

**Tokenisering:**

CST's tokeniser forbereder tekst til viderebehandling med andre programmer. Ofte kræver programmer til tekstanalyse nemlig at inputteksten er gemt på en helt bestemt måde, hvilket næsten altid indebærer at tekstfilen skal være fri for styretegn til layout og typografi, skal have et bestemt tegnsæt, samt at teksten er opdelt i tokens og eventuelt også i segmenter (sætninger, overskrifter og listepunkter). Ved opdelingen af teksten i tokens (tokenisering) adskiller man ord, tal, interpunktionstegn mm. med et mellemrum og samler evt. flerordsforbindelser (fx i forhold til) til et token.

Mange tekster foreligger ikke i det ønskede flade tekstformat og skal først konverteres. Eksempler er PDF-filer, HTML-filer og Microsoft Word-dokumenter. Konvertering fra tekst-med-layout til flad tekst er meget mere end fjernelse af de tegn der styrer layoutet og typografien. I layoutet og typografien ligger værdifulde oplysninger om både tokens og sætninger. Fx slutter overskrifter normalt ikke med punktum. Det er udelukkende overgangen fra stor, fed skrift til normal skrift der kan fortælle hvor overskriften slutter. Og punkter ("bullets") i en liste kan i nogle tilfælde kun kendes ved at man kigger på både typografi og placering.

CST's tokeniserings- og segmenteringsprogram er udviklet til at konvertere fra RTF (Rich Text Format) til det ønskede flade tekstformat. Vi har valgt RTF fordi det er veldokumenteret og fordi dokumenter som er skabt i andre formater, såsom PDF, HTML og DOC, nemt kan konverteres til RTF uden tab af afgørende layout-informationer.

Programmet kan i øvrigt også tokenisere og segmentere flade tekster, men kvaliteten er typisk lidt mindre på grund af de manglende layout-oplysninger.

### 1.3.1 Kvalitetskontrol af dokumentrengøring

Kvalitetskontrollen af dokumentrengøring og tokenisering må nødvendigvis omfatte manuel evaluering af stikprøver. Der vil være begrænset tid til denne kontrol, men den er ikke desto mindre vigtig fordi fejl på dette niveau vil slå igennem i al følgende lingvistisk annotation.

# 2. Tekniske løsninger: Korpusopmærkning

## 2.1 Dokumentannotation (annotation med metadata)

I juni 2009 blev der fastlagt et fælles metadataformat for alle korpustekster i WP 2 (se http://korpus.dsl.dk/clarin/corpus-doc/text-header.pdf). Alle tekster i arbejdspakken skal altså udrustes med headere som overholder dette format. Headerformatet overholder TEI P5, men denne standard fastlægger imidlertid kun struktur og ikke indhold, og dermed kan man fx selv frit vælge sin egen emnetaksonomi og anvende dennes deskriptorer (fx i elementet profileDesc/textClass/catRef[[3]](#footnote-3)). Værdinventaret for de forskellige elementer i strukturen udgør således en foreningsmængde af alles behov.

I WP 2.2 vil et helt centralt metadataelement være "emnekategori". Dette element definerer hvilket subkorpus (og dermed fagområde) det pågældende dokument tilhører. Som emnetaksonomi anvendes hjemmestrikkede betegnelser, da DK5 eksempelvis ikke har den rette detaljeringsgrad og snarere afspejler hvad der findes på bibliotekshylderne end hvad der findes i verden så at sige. DK5 er dermed mere oplagt at anvende i WP 2.1. Udover de metadata WP 2.2 deler med WP 2.1 (fx administrative oplysninger) vil WP 2.2 i særlig grad udnytte det udvidede værdiinventar for de følgende elementer:

* Kval\_dansk: decimaltal i intervallet 0-1 (hvis lavt => lavt indhold af engelsk)
* Kval\_lix: heltal
* Termtæthed: //fileDesc/extent/num[n=terms] (NB: sat i forhold til antal tokens!)
* Scannet: //fileDesc/titleStmt/respStmt/name/note[method]
* Oversat: //sourceDesc/biblStruct/analytic/respStmt/resp[Translated by]
* Kommunikationstype[[4]](#footnote-4): expert-expert | expert-advanced | expert-basic | advanced-basic
* Tekstudstrækning: //ecodingDesc/samplingDeclaration[CTB sample | version | excerpt]

**Tekniske løsninger**

Metadataelementer for korpustekster kan enten være invariante for arbejdspakken, invariante per leverandør eller variante per tekst. Mulighederne eksemplificeres via nedenstående punktopstilling.

* Helt invariante metadata (fx //projectDesc/projectIdentifier)
* Metadata der er invariante per leverandør (fx domæne)
* Metadata der varierer per tekst
  + men kan genereres automatisk (fx id, textFileName, textUrl, antal tokens, antal types, indsamlingstidspunkt, leverandørid)
  + og må indtastes manuelt (fx hele //sourceDesc/biblStruct)

**Headergeneratoren**

I løbet af foråret 2009 har WP 2.2 udviklet en såkaldt headergenerator som netop tager højde for de variante og invariante metadataelementer og forsøger at minimere det tidsforbrug der uvægerligt vil være i forbindelse med udfyldelsen og lagringen af tekstheadere i arbejdspakken. Værktøjet (en webapplikation udviklet i Java) er næsten færdigudviklet og er tilgængeligt på <http://192.38.108.131:8080/clarinTools/>.

Headergeneratoren er kort fortalt en indtastningsformular med et antal felter der repræsenterer alle de variante metadataelementer. Hvert felt er kædet sammen med et XPath-udtryk som identificerer positionen af det pågældende XML-element i headerstrukturen. Hvert felt er endvidere kædet sammen med en datamodel (en "java managed bean"), hvilket muliggør automatisk konvertering og validering af feltværdierne inden brugeren får lov at submitte sin header. Desuden sikrer datamodellen og frameworket JSF at grænsefladens tilstand (dvs. de indtastede værdier) bevares i hver HTTP session, hvilket gerne skulle minimere indtastningsbehovet per tekst.

Når en header er indtastet og valideret, får brugeren mulighed for at anvende et default headerskelet med alle de invariante metadata eller anvende sit eget headerskelet. Endelig skal brugeren uploade den tilhørende tekstfil (i det oprindelige format), hvis denne ikke allerede er uploadet.

Header og tekst gemmes derpå i to separate databasetabeller, men er kædet sammen via en id. Det er så tanken at teksterne efterfølgende kan trækkes ud af databasen, konverteres til standardtekstformat (se afsnit 3.1), kædes sammen med deres tilhørende header og gemmes i den fælles tekstbank (XML-databasen, eXist).

Kvalitetskontrol af såvel automatisk genererede metadataelementer som manuelt indtastede metadataelementer vil foregå ved manuel stikprøvekontrol.

## 2.2 **Tekstannotation** (lingvistisk annotation)

Tekstannotation vil som sagt foregå ved at der genereres span-elementer (se http://korpus.dsl.dk/clarin/corpus-doc/text-format.pdf) der refererer til w-elementer i standardtekstformatet vha. XML-id. Dermed kommer al lingvistisk annotation til at ligge i separate lag "ved siden af” basiskorpus. Dette har mange fordele, blandt andet har man derved mulighed for flere forskellige løsninger for annotation inden for samme annotationstype (fx PoS-tagging).

Ingen af de nedenfor nævnte værktøjer kan pt. tage udgangspunkt i det definerede standardtekstformat, og der kræves en del omprogrammering før dette kan lade sig gøre.

### 2.2.1 PoS-tagging

Vi forventer at korpusset minimum annoteres med CST’s version af Eric Brills PoS-tagger[[5]](#footnote-5). Hvis det er muligt, tagges korpus også med andre frit tilgængelige PoS-taggere.

POS-tags kan bruges i viderebehandling af teksten fx til at udtrække bestemte ordklasser (alle finitte verber, alle substantiver mm.), til at afgøre hvilken ordklasse et givent ord tilhører i en given position (fx jeg *løber* (verbum) vs. en *løber* (substantiv), eller til at gruppere ordklasser i syntagmer

CST's POS-tagger er en udvidet udgave af Brills PoS-tagger, med tilføjelser til bedre håndtering af ord med store bogstaver i fx overskrifter.

Taggeren er trænet på Parole-korpusset[[6]](#footnote-6) så de regler den bruger til at beregne ordklassen for nye ord eller for homografer, afspejler sammensætningen og sprogbruget i Parole-korpusset. Under optimale forhold kan man opnå et resultat på op til 97% korrekt tagging.

### 2.2.2 Lemmatisering

Lemmatisering er et andet fundamentalt lag i tekstannotationen som muliggør mere sofistikerede søgninger i korpusset.

CST’s lemmatiser beregner grundformen/opslagsformen af ord ud fra en række regler og en ordbog der begge afspejler forholdet mellem ordformer og grundformer. Reglerne er for den danske version af lemmatiseren genereret på baggrund af [STO-ordbasen](http://cst.dk/sto/index.html)[[7]](#footnote-7).

På grund af STO-basens omfang og kvalitet beregnes lemmaet med 94 -98 % nøjagtighed, afhængigt af om inputtet er POS-tagget eller ej. Hvis inputtet ikke er POS-tagget, kan lemmatiseren fx ikke skelne mellem *dør* som udsagnsord og *dør* som navneord.

Lemmatisering er anvendelig til mange forskellige formål, fx til at samle alle former af samme lemma, beregne ordform- og lemmafrekvenser og vise hvilke former af lemmaet der forekommer i teksten, hvilket er første skridt i indeksering af en tekst. Lemmatiseren kan også markere hvilke ord der findes eller ikke findes i ordbogen (STO eller brugerens egne), hvilket er nyttigt til identifikation af nye termer.

### 2.2.3 Term-tagging

I modsætning til lemmatisering og pos-tagging er term-tagging et mere kontroversielt lag i tekstannotationen. Der er forholdsvis stor konsensus om hvad et lemma er, og hvad en ordklasse, fx et adjektiv, er, men der er en væsentligt mindre grad af konsensus om hvad en term er, om der er nogen formel forskel på ord og termer (se fx Kageura 2002), og hvori forskellen på fagsproglige begreber og almensproglige kategorier består (se fx Temmerman, 2000 og Spang-Hanssen, 1983).

Dette har imidlertid ikke forhindret den datamatiske terminologi i at udvikle en række henholdsvis statistiske og lingvistiske metoder til at genkende termer i tekst. Der findes et hav af statistiske mål for "termhood", eksempelvis TFIDF, "weirdness" (Ahmad, 1993) samt en række associationsmål[[8]](#footnote-8) der også kan anvendes til at fremfinde kollokationer. De fleste mål sammenholder en termkandidats frekvens i analysekorpusset (den pågældende tekst) med dens frekvens i et større referencekorpus og ser om forholdet afviger fra det statistisk forventelige.

Hvad end man anvender statistiske, lingvistiske eller hybride metoder til at genkende termer, bør man overveje om man også vil annotere flerordstermer. I rent statistiske tilgange vil man i givet fald skulle lægge sig fast på et tal, fx bi eller trigrammer. I lingvistiske tilgange vil man skulle lægge sig fast på et morfosyntaktisk mønster (typisk en NP-skabelon). Fordelen ved dansk er at sammensatte substantiver sammenskrives, og man vil derfor sandsynligvis kunne komme ganske langt med unigrambaserede teknikker.

WP2.2 har endnu ikke besluttet hvilke metoder der skal bruges til term-tagging; men vi forventer at eksperimentere med minimum CST's *keyword extractor*, om muligt også med andre statistiske metoder som er blevet anvendt på eksempelvis Korpus 2000 (jvf. Asmussen, 2006), så som *log-likelihood*.

CST's *keyword extractor* udtrækker 20 keywords der karakteriserer en tekst. Det gøres ved at tekstens ord sammenlignes med ord i Berlingske Tidende årgang 1999. Ord i teksten antages for at være karakteriserende for teksten hvis de kun forekommer i forholdsvis få artikler i Berlingske Tidende, dvs. hvis de ikke blot er almindelige frekvente ord.

Den undersøgte tekst PoS-tagges og lemmatiseres. Derefter udregnes substantivernes relative frekvens vha. den velkendte vægtningsalgoritme TF\*IDF[[9]](#footnote-9) som kombinerer termfrekvens (TF) og den omvendte dokumentfrekvens (IDF). Funktionen er i stand til at udskille de termer der karakteriserer et individuelt dokument, fra termer der karakteriserer en hel dokumentsamling eller almensprog:

TF\*IDF = log10((*n*/*df*)\**tf*)

hvor *n* er antallet af dokumenter, *df* er antallet af dokumenter hvori termen forekommer, og *tf* er termens frekvens i det undersøgte dokument.

De 20 keywords er tænkt som en liste hvorfra man kan vælge de ord man synes karakteriserer en tekst bedst. Men man kan fx også tilføje dem automatisk til en hjemmesides metatags. Det vil selvfølgelig være muligt at udskifte referencekorpusset så man ikke sammenligner med almensprog i Berlingske Tidende; men fx i Korpus2000.

Vi vil her kort nævne to andre, meget populære statistiske metoder til term-tagging, nemlig odds-ratio (log-odds) og log-likelihood. Disse metoder tager ikke hensyn til dokumentgrænser, men sammenholder alene ordformers frekvenser i to tekstsamlinger baseret på et såkaldt "contingency table"[[10]](#footnote-10):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ord=X** | **ord≠X** |  |  |  | **ord=X** | **ord≠X** |
| **korpus=Y** | O11 | O12 | =R1 |  | **korpus=Y** |  |  |
| **korpus≠Y** | O21 | O22 | =R2 |  | **korpus≠Y** |  |  |
|  | =C1 | =C2 | =N |  |  |  |  |

Forskellen på log-odds og log-likelihood er at førstnævnte alene analyserer ordformens observerede frekvenser (tabellens venstre side), mens log-likelihood også tager ordformens forventede frekvenser i betragtning (tabellens højre side). Dette har den helt konkrete konsekvens at log-odds tillægger sjældne begivenheder stor betydning (dvs. identificerer de virkeligt specielle ordformer i en tekstsamling), mens log-likelihood er mere konservativ og ignorerer de sjældnere termkandidater. Begge metoder har deres fordele, og man vil med fordel kunne anvende en kombination.

****

****

DSN har udviklet en termgenkenderprototype der anvender ovenstående odds-ratio, men som endnu ikke kan tage standardtekstformat som input og generere spans der overholder specifikationerne som output. Termgenkenderen udvikles løbende.

### 2.2.4 Kvalitetskontrol af tekstannotation

Kvalitetskontrol af den automatiske PoS-tagging og lemmatisering involverer manuel stikprøvekontrol. Wp2.2 har endnu ikke defineret hvordan.

Fælles for samtlige tilgange til term-tagging er at de alene identificerer term**kandidater**, og korpusbrugerne vil således uvægerligt spotte et antal fejlagtige oplysninger i dette lag af annotationen. Da der næppe er resurser til omfattende manuel kvalitetskontrol af term-taggeren[[11]](#footnote-11), gælder det om at vælge et ret konservativt statistisk mål for "termhood" som har en høj præcision, men måske en lavere ’recall’. Man vil sandsynligvis med fordel kunne kombinere et antal lukkede vokabularer (eksempelvis fra Euroterm, Agrovoc eller mere generelt fra wikipedia-artikler) med et statistisk mål for at sikre en højere præcision.

1. http://listento.jaketolbert.com/?page\_id=1661 [↑](#footnote-ref-1)
2. http://blog.antenna.co.jp/PDFTool\_en/archives/2006/02/ [↑](#footnote-ref-2)
3. http://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-profileDesc.html [↑](#footnote-ref-3)
4. //profileDesc/textDesc/interaction/note[type=interactRole] [↑](#footnote-ref-4)
5. http://cst.dk/download/index.html#tagger [↑](#footnote-ref-5)
6. http://korpus.dsl.dk/paroledoc\_dk.pdf [↑](#footnote-ref-6)
7. http://cst.dk/sto/index.html [↑](#footnote-ref-7)
8. se fx http://collocations.de/AM/index.html [↑](#footnote-ref-8)
9. Se fx Manning & Schütze 1999, p.543 [↑](#footnote-ref-9)
10. tabellen er taget fra http://collocations.de/AM/index.html [↑](#footnote-ref-10)
11. dette vil ideelt set fordre hvervning af eksterne domæneeksperter [↑](#footnote-ref-11)