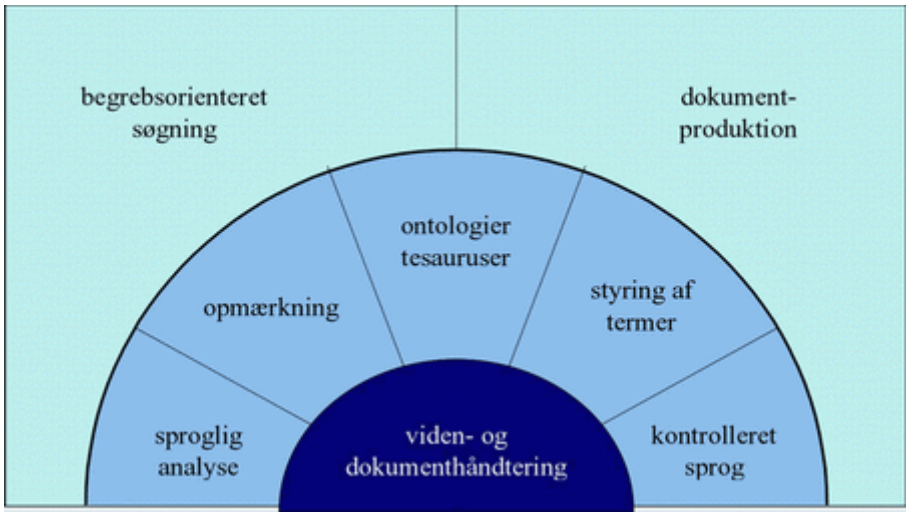


Automatisk analyse af Zaccos og Ankiros tekstmateriale

Bart Jongejan, Bolette S. Pedersen, Costanza Navarretta

VID-rapport nr. 3

Center for Sprogteknologi

2004

Rapporten kan fås ved henvendelse til CST, cst@cst.dk, eller hentes fra CST's hjemmeside www.cst.dk.

VID-projektet er støttet af Center for Informationsteknologi (nu overgået til Forskningsstyrelsen).

Om VID: Viden- og Dokumenthåndtering med sprogteknologi

Der er et udtalt behov hos danske virksomheder for at kunne supplere deres eksisterende sproglige kompetence og viden med sprogteknologiske IT-værktøjer og metoder som dels kan støtte medarbejderne, dels forankre viden og processer i virksomhedens IT-systemer, dels danne grundlag for den udvikling der kræves hvis virksomhederne skal overleve og vokse i den stadigt mere globaliserede økonomi.

VID-projektet er et forsknings- og udviklingsprojekt der har til formål at udforske de forskellige muligheder som sprogteknologi frembyder inden for informationssøgning og dokumentproduktion, og at understøtte de deltagende virksomheder i at udvikle værktøjer til bedre udnyttelse af egen viden, samt til bedre og mere effektiv produktion af dokumentation, herunder flersproget dokumentation. Foruden CST omfatter projektet på den ene side virksomhederne Bang & Olufsen A/S, Zacco A/S og Nordea A/S, som i dette projekt udgør teknologiens brugere, på den anden Navigo Systems A/S og Ankiro, som er teknologiproducenter. Projektet omfatter følgende forskningsopgaver:

- analyse af de tekstuelle data virksomhederne skal kunne håndtere for at kunne fastlægge tesauruser/ontologier for de relevante semantiske domæner, undersøgelse af den bedst egnede formalisme/teknologi til at udtrykke disse;

- afdækning og videreudvikling af sprogteknologiske komponenter til brug for automatisk tekstklassifikation og begrebsorienteret informationssøgning, indbefattende tilpasning af sprogteknologiske 'basismoduler' til opmærkning af tekst;

- udforskning af flertydighed i tekstuelle data som kan vanskeliggøre informationssøgning; ligeledes den omvendte problematik: at samme indhold kan udformes forskelligt rent sprogligt og derfor kan være svær at fremfinde i store datamængder;

- forskning inden for kontrolleret sprog - også set i et flersproget perspektiv - til brug for dokumentproduktion; herunder analyse af den sprogstil og tone som virksomhederne ønsker at anvende, samt opstilling af modeller for dette sprog;

- undersøgelse af hvilke sprogteknologiske metoder der kan anvendes til denne kvalitetssikring af dokumentproduktionen i form af f.eks. termstyring og grammatikkontrol.

Projektet er støttet af Center for IT-forskning og løber i perioden 2003-2004.

Indhold

1 Indledning

Denne VID-rapport omhandler automatisk analyse af det tekstmateriale som CST har modtaget i relation til delprojektet om ontologi og søgning fra to af de deltagende virksomheder i VID: Zacco A/S og Ankiro. Som nævnt i det indledende afsnit om VID-projektet udgør Zacco som IPR-virksomhed en såkaldt bruger i projektet, dvs. at de har et konkret sprogteknologisk behov: virksomheden ønsker at opbygge en videnbase indenfor sagsbehandling af patent- og varemærkesager således at dokumentproduktion og –vedligeholdelse kan blive mere effektiv, og således at der kommer et bedre videnflow i virksomheden. Til dette formål har de bl.a. brug for at få afdækket deres fagterminologi og opstillet denne i en struktureret form, gerne i form af en ontologi. Zaccos standarddokumenter til sagsbehandling danner udgangspunkt for dette arbejde.

Ankiro er derimod en teknologiudviklervirksomhed som bl.a. arbejder med at udvikle intelligente søgemaskiner. Et af deres mål som deltagere i VID-projektet er at undersøge nogle sprogteknologiske metoder i forbindelse med nogle meget specifikke, ontologisk relaterede problemer så som hvordan man automatisk kan identificere synonymer og synonyme udtryk; altså udtryk der refererer til samme begreb.

I denne rapport fokuseres der på en meget specifik problemstilling i relation til dette, nemlig *sammensatte ord* og søgefunktionaliteten i relation til dette. Det er et kendt problem at det kan være vanskeligt at søge på sammensatte ord fordi disse typisk har mange alternative udtryksformer; udtryksformer som har samme semantiske indhold. Om man siger *byrådsmedlem* eller *medlem af byrådet*, *husholdningsaffald* eller *affald fra husholdninger* er altså mere eller mindre underordnet; udtrykkene er parvis synonyme. Men søger man på *byrådsmedlem* vil man med en almindelig søgemaskine kun få fremfundet tekster hvor lige præcis det sammensatte ord forekommer, og det betyder at søgemaskinens recall bliver relativt lavt[1]; der er altså højst sandsynligt mange relevante tekster der ikke bliver fundet. Virksomheden er derfor interesseret i at få undersøgt hvilke sprogteknologiske analyser der kan være med til at bestemme om hits med 'splittede'[2] sammensatte ord er gode eller dårlige. De har derfor leveret to korpora bestående af en række tekstudsnit i form af søgehits som er fremkommet ved at man har splittet sammensatte ord og søgt på disse. I stedet for at søge på *byrådsmedlem* har man altså søgt på *byråd* og *medlem*. En af de grundlæggende hypoteser for de sprogteknologiske analyser har været at hits hvor begge søgeord var at finde inden for den samme navnefrase, sandsynligvis var gode hits som skulle prioriteres relativt højt.

Rapporten består af 4 kapitler udover indledningen. Kapitel 2 omhandler faktuelle data om teksterne, mens kapitel 3 omhandler den automatiske processering af teksterne som er foregået på CST (bl.a. tokenisering, tagging og parsing). I kapitel 4 berettes om den

[1] Hvis en given database antages at indeholde i alt 50 dokumenter, der kan karakteriseres som værende relevante i forhold til en forespørgsel, og samme forespørgsel fremfinder alle 50 dokumenter, så har den en *recall* på 1. Hvis der kun fremfindes 10 af de 50 relevante dokumenter, så er *recall* på 10/50=0,2, hvilket omvendt betyder 0,8 (80%) af de relevante dokumenter ikke blev fundet, og derfor stadigvæk er ukendte for brugeren (http://www.pce-web.dk/search/size.htm).

[2] *Medlem af byrådet* opfattes altså i denne sammenhæng som 'splittet'.

semiautomatiske produktion af termlister, mens vi i kapitel 5 rapporterer om arbejdet med identifikation af tekststumper i Zaccos materiale. I kapitel 6 konkluderer vi på det udførte arbejde og diskuterer hvilke manuelle analyser af data som forestår.

2 Facts om teksterne

2.1 Zaccos tekster

Zaccos tekster er en slags halvfabrikater. Noget af teksterne vil bogstaveligt stå i en endelig tekst, mens andre dele af teksterne snarere er forfatterhjælp, som kan tage mange former:

1) Som inputfelt (En grå boks uden eller med tekst, som forsvinder ved input fra tastaturet og erstattes af den indtastede tekst.)

    a. Et felt med forklarende tekst af feltens indhold, fx

    Dato: XX måned XXXX

    b. Et tomt felt uden forklarende tekst, fx

    I henhold til Deres instruktioner af [ ] har vi den [ ] indleveret ovennævnte patentansøgning med prioritet fra [ ] patentansøgning af [ ].

    c. Et tomt felt med forklarende tekst som kommentar, fx

    til brugen af ovennævnte varemærke i Danmark [ ](*angiv evt. mere begrænset geografisk område*) i forbindelse med [ ](*angiv varer/tjenesteydelser eller all goods/services covered by the registration*).

2) Som kommentar

    a. I parenteser, fx

    (Vores kommentarer)

    Vi imødeser Deres snarlige kommentarer og instruktioner inden fristens udløb den

    Bemærk at tekst i parenteser ikke nødvendigvis er kommentar.

    b. Ikke i parenteser, fx
    - Vedlæg overdragelsesdokument(er) til underskrivning eller ryk for oplysninger om ansøger(e) og opfinder(e)
    - *Vedlæg andre eventuelle bilag*

3) Som et valg

    a. Mellem alternative termer adskilt med skråstreg

i. Uden blanktegn omkring skråstreg

sagsøger/sagsøgte

ii. Med blanktegn omkring skråstreg

overdragelsen / navneændringen / adresseændringen

b. Mellem alternative dele af termer adskilt med skråstreg

i. Med bindestreg

Navne-/adresseændring i EF-varemærkeansøgning/-registrering

ii. Uden bindestreg

at notere navne/adresseændringen i EF-varemærkeregisteret

c. Ved hjælp af en valgfri tilføjelse, fx (bemærk at e'et i "(e)" ikke er valgfrit, medmindre man erstatter "følgende" med "en"!):

fra følgende national(e) registrering(er) i forbindelse med ovennævnte EF-varemærke

Nogle gange er det svært for en udenforstående at afgøre om tekst skal tages bogstaveligt eller opfattes som forfatterhjælp, jvf. denne formular som måske kun kræver indtastning af felterne, mens teksten, parenteser inklusive, skal forblive uændret:

Ordrebekræftelse - Dansk grundansøgning

...

Indleveringsdata

Ansøger(e):

Navn:

Adresse:

Oplysninger vedrørende ansøger(e) (navn og adresse) bedes fremsendt hurtigst muligt

…

Bilag

Vedlagte overdragelseserklæring(er) bedes underskrevet som anført og returnet

til undertegnede hurtigst muligt og senest inden indleveringen af ansøgningen

Overdragelseserklæring(er) til underskrivelse vil blive fremsendt, når vi har modtaget de nødvendige oplysninger fra Dem

…

2.1.1 Zaccos standardtekster

Denne samling omfatter 202 Word (DOC) filer i 25 undermapper af 9 mapper og er modtaget 2003.06.11. I alt er der 47.341 ord. Sproget er dansk.

Der var problemer ved modtagelse pr. e-mail:

- Filerne blev modtaget i ca. 30 e-mails. Nogle forsendelser blev dog kaldt tilbage.
- Dokumentnavnene indeholdt ingen mappespecificering. Da det viste sig at mange dokumenter havde samme navn, var det nødvendigt manuelt at oprette en mappestruktur for at undgå at dokumenter overskrev hinanden.

Disse problemer kunne have været undgået ved at Zippe alle dokumenter i én zipfil før forsendelsen. Et sådant Zip-arkiv af alle dokumenter fylder 920 KB, mens dokumenterne tilsammen fylder 5,63 MB.

2.1.2 Zaccos varemærketekster

Denne pulje består af 76 filer i én mappe, modtaget 2003.07.07.

Format: Microsoft Word skabelon (DOT).

Problemer ved modtagelse:

1) Én fil optrådte to gange (AS400 VEJLEDNING.dot) i forsendelsen. Der kunne være tale om to forskellige filer. (Se forrige afsnit.)

2) Det var umuligt (eller i hvert fald svært) at konvertere skabelonerne til RTF filer.

Zacco sendte os et nyt sæt filer, denne gang i DOC format. Der var igen problemer ved modtagelsen.

1) Filnavnene på mange af dokumenterne var forvansket af mailsystemet. Selve mail-filen havde et format der indeholdt fejl på grund af dokumenternes filnavne. Først efter manuel redigering af mail-filen kunne dokumenterne pakkes ud.

2) To dokumenter optrådte to gange i forsendelsen: ”CTM - Rapport klient vedr notering af navne-adresseændring” og ” Konflikt - Told og skat - anm om 10 dages suspension”. Se tidligere problembeskrivelse.

3) Forsendelsen indeholdt ikke alle tekster som var med i den første forsendelse: AS400 VEJLEDNING manglede.

Efter fjernelse af dubletter indeholdt det anden sæt 10.911 ord i 74 filer. Sproget er dansk.

2.2 Ankiros tekster

Vi har modtaget 3 filer:

1) vid_data.zip 2003.08.15

   Denne fil indeholder en flad tekstfil "vid_data.txt" på 2.681 KB

2) vid_data1.zip 2003.09.19

   Denne fil indeholder en flad tekstfil "vid_data1.txt" på 15.155 KB

3) vid_data1.zip 2004.02.11

   Denne fil indeholder en flad tekstfil "vid_data1.txt" på 9.297 KB

Filerne har samme opbygning. Hver fil indeholder et antal søgespecifikationer, efterfulgt af hittene for hver søgning. Den sidste version af vid_data1.zip indeholder resultaterne for 1044 søgninger og i alt 32664 resultater. En søgespecifikation ser således ud:

```
<WORD>dataadministration   administration            data</WORD><COUNT>363</COUNT>
```

Resultaterne vises i to kolonner, ét søgeresultat per række:

```
1364839      Liste over <T9>data</T9> grupper. DataGroupList (<T9>Administration</T9>)
Hjælp til nr. 11.
1364840      Redigering af specifik <T9>data</T9> gruppe. DataGroupEdit
(<T9>Administration</T9>) Hjælp til nr. 12.
1166381      sektor. Den omfattende anvendelse af IT i kommunernes
<T9>administration</T9> tilvejebringer masser af <T9>data</T9> og informationer - men
systemerne er ofte lavet med
```

Normalt indeholder et søgeresultat to markerede ord. Markeringen sker ved hjælp af sgml tags <Tn> … </Tn>, n = 1..9. Jo højere n, jo "bedre" er resultatet. Et søgeresultat er en stump tekst som indeholder søgeordets komponenter eller synonymer deraf. Stumpen er valgt således at teksten før det første markerede ord indeholder cirka lige så mange ord som teksten efter det andet markerede ord. Med andre ord: resultatet er centreret omkring det markerede ordpar. Nogle resultater indeholder færre eller flere end to markerede ord.

Forskellene mellem vid_data.txt og vid_data1.txt er, at vid_data1.txt er meget større, at søgeresultaterne indeholder mere tekst og at der ikke ekspanderes til synonymer. I den første version af vid_data1.txt var der nogle fejl: de taggede ord var ikke altid centreret

og mange resultater indeholdt ikke markerede ord. Det viste sig at der var lavet fejl ved genereringen af vid_data1.txt, hvorfor vi senere modtog en forbedret udgave.

3 Processering af teksterne

3.1 Zaccos tekster

For at få overblik over Zaccos tekster er det nødvendigt at have et repræsentativt tekstkorpus til rådighed. Ved hjælp af et korpus og et passende søgeprogram kan man hurtigt og nøjagtigt fastslå hvordan en given term bruges i Zaccos eksisterende dokumenter, fx hvor ofte termen bruges, og om den bruges i mere end én betydning.

Zacco forsynede CST med et stort antal dansksprogede tekster. Disse tekster blev konverteret til 'flade' tekster (dokumenter kun indeholdende bogstaver, mellemrum og læsetegn, blottet for afsnits- og tekstformatering og billeder). De flade tekster blev behandlet af et program som bestemmer ordklassen for hvert ord. Tekst og ordklassebestemmelserne blev derefter lagt sammen i et tekstkorpus som er tilgængelig for diverse søgeprogrammer. Ydermere brugte vi førnævnte analyseværktøj til at lave rå analyser af teksterne.

3.1.1 Konvertering til flad tekst

For at kunne bestemme et ords ordklasse er det nødvendigt at arbejde på både ord- og sætningsniveau. For et computerprogram er det ikke altid ligetil at bestemme hvor et ord starter og slutter. Problemet er endnu større ved bestemmelsen af start og slut på en sætning. Normalt kan et program nøjes med at søge efter blanktegn og punktummer, men det er ikke altid nok. Fx afslutter et punktum normalt en sætning, men det kan også være en del af en forkortelse. Og nogle 'sætninger' slutter ikke med et punktum, fx overskrifter. Ved konvertering fra et dokument med lay-out oplysninger til et dokument uden disse er det derfor vigtigt at udnytte lay-out oplysninger til bestemmelse af ord- og sætningsgrænserne, før lay-out oplysningerne bliver fjernet. Fx kan overskrifter normalt kendes ved at de er sat i en anden skrifttype eller skriftstørrelse end brødteksten. Særdeles svære er 'bullets'. Disse er symboler hvorom det kan siges at de står forrest på en linje og gentages mindst én gang, men udformningen kan variere fra specielle tegn til kombinationer af tegn som også bruges andre steder, fx en bindestreg efterfulgt af et blanktegn. Og det står hen i det uvisse om 'bullets' skal bibeholdes i den flade tekst, og hvis ja, om de skal repræsenteres på en ensartet måde.

Det viste sig at standardapplikationer som kan konvertere fra DOC format til flad tekst (TXT), ikke selv er i stand til at opdele teksten i sætninger. Ej heller var det muligt, på grund af tabt lay-outinformation, at opdele teksterne i sætninger ved en efterbehandling. Derfor udviklede vi et program som kan konvertere RTF-tekster til flad tekst - med én sætning per linje. RTF er et format som de anvendte dokumentformater kan konverteres til med bibeholdelse af det meste af relevant lay-out. Samtidig er RTF en åben standard, om end med en komplicerende udviklingshistorie på bagen. Konverteringsprogrammet løser problemet med punktummer som afslutter en forkortelse ved at slå kandidat-forkortelser op i en liste af kendte 'forkortelser-med-punktum'. Programmet er også i stand til at spotte flerordsenheder ved hjælp af opslag i en ekstern liste.

3.1.2 Ordklassebestemmelse

Vi brugte et offentligt tilgængeligt værktøj til at bestemme ordenes ordklasse, Eric Brills POS-tagger[3] (POS = Part Of Speech). Vi har ændret programmet for bedre at håndtere overskrifter og ord som starter med stort bogstav. POS-taggeren tager flad tekst som input, med én sætning per linje og alle ord og tegn adskilt med blanktegn.

Til tagging af Zaccos tekster brugte vi denne tagger med CST's egne danske lingvistiske resurser.[4]

Taggeren fungerer forholdsvis dårligt med Zaccos tekster på grund af de mange felter, indføjede kommentarer og valgmuligheder. Vi har ikke forsøgt at lave en "intelligent" behandling af disse støjkilder. Problemet er at vi ikke ved hvad man skal stille op med fx "EF-varemærkeansøgning/-registrering" for at tilfredsstille både taggeren, parseren og de værktøjer som skal kigge nøjere på de anvendte termer. Og hvad kan man gøre ved en ugrammatisk sætning som følgende ?

I henhold til Deres instruktioner af [redacted] har vi den [redacted] indleveret ovennævnte patentansøgning med prioritet fra [redacted] patentansøgning af [redacted].

Det kræver en mere avanceret tagger end Brill's at tage højde for huller og valgmuligheder i teksten. De mest frekvente problemer med den automatiske processering skyldtes følgende fænomener:

- Filnavne (som også indgår i dokumenterne) indeholder blanktegn, parenteser og kommaer samt danske bogstaver. De første tre typer tegn må ikke bruges i filnavne i UNIX. Danske bogstaver kan anvendes, men kræver tilpasning af visse UNIX-programmer.

- Dokumenterne indeholdt en del engelske ord og termer som bliver behandlet som danske ord.

- Forkortelser af typen "dokument(er)" og "ord-" i "ord- og termbase", samt alternationer som "rapport/dokument/brev" bliver behandlet som enkelte tokens.

- Kun nogle forkortelser, som blev anvendt i dokumenterne, bliver genkendt i den anvendte tokeniser. Det samme gælder flerordstermer (både på engelsk og på dansk). Tegn "-" blev nogle gange genkendt som tegn, andre gange som del af ord. Sekvenser af flere punktummer "." bliver ikke genkendt på en korrekt måde.

[3] http://www.cs.jhu.edu/~brill/RBT1_14.tar.Z
[4] http://cst.dk/tagger/rapport/index.html

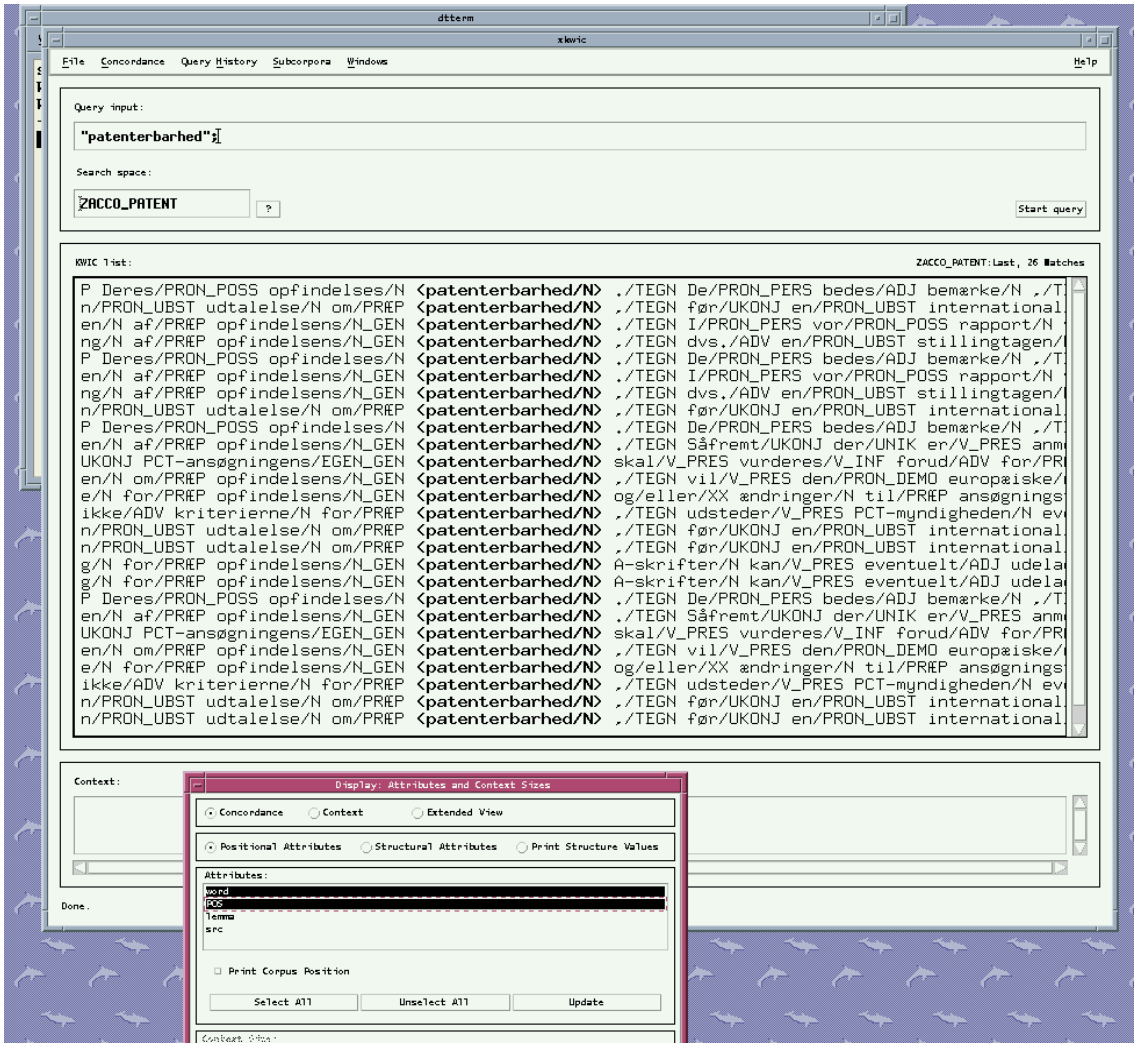
- De oprindelige word-dokumenter indeholdt en del sektionsnavne, tabeller og punktopstillinger som, konverteret til tekstfiler, er resulteret i usædvanlige rækkefølger af ord. Da taggeren er trænet på regelmæssige sætninger, behandles disse ordrækker som sætninger. Dette resulterer i opmærkningen med forkerte tags.

- Ord der startede med store bogstaver bliver i de fleste tilfælde tagget som proprier, mens nogle proprier ikke bliver genkendt korrekt.

- Dokumenterne indeholder en del imperativer, som ikke bliver genkendt korrekt af taggeren (trænet på andre typer konstruktioner).

3.1.3 Opbygning af korpusser

De flade, POS-taggede tekster er blevet kopieret efter hinanden i store filer, én fil for standardteksterne og én fil for trademarkteksterne. Disse store filer er blevet lagt ind i et søgesystem, XKWIC. Programmet gør det muligt at søge på ord, men også avancerede søgestrenge, indeholdende (dele af) ord, jokertegn og ordklasser, er mulige. Resultatet af søgningen vises som en konkordans, hvilket vil sige at man får en liste af alle tekstdele hvori søgemønstret forekommer, med søgemønstret i midten af resultatvinduet, omgivet af konteksten af ordet. Men kan vælge at få vist hvilken ordklasse ordene tilhører og fra hvilke dokumenter linjerne stammer.



Via internettet – med brug af passwords – kan man søge i Zaccos korpusser med en forenklet udgave af XKWIC[5].

3.2 Ankiros tekster

Formålet med processeringen af Ankiros tekster var at finde ud af hvordan de markerede ord forholdt sig til hinanden, set fra en syntaktisk synsvinkel. Især ville vi undersøge om hits med begge søgeord i samme NP generelt var bedre hits end de øvrige.

For at undersøge dette var det nødvendigt at POS-tagge og parse søgeresultaterne. Vi stødte på følgende komplikationer:

[5] http://cst.dk/cgi-bin/defisto/

1) Søgeresultaterne bestod normalt ikke af hele sætninger, men af en, fra en syntaktisk synsvinkel set, vilkårlig sekvens af *n* ord, hvor *n* er et antal som var næsten ens for næsten alle søgeresultater. (20 for vid_data.txt, ca 40 for vid_data1.txt). Både POS-taggeren og parseren kunne have problemer med sætninger som manglede starten og/eller slutningen.

2) <Tn> og </Tn> markeringerne skulle ikke tagges. Da Brill's tagger ikke kan lade disse markeringer være uberørt, men prøver at tagge dem, var det nødvendigt at fjerne markeringerne, tagge, parse teksten, og genindsætte markeringerne. Det sidste var det svære.

3) Tallene i den første kolonne, som ellers ville have været en fin identifikator på et søgeresultat, skulle ikke tagges og måtte derfor fjernes.

4) Efter at <Tn>- og </Tn>-markeringerne og tallene var fjernet og før taggeren skulle processere teksten, var det nødvendigt at tokenisere teksten og opbryde teksten således at to sætninger ikke stod på samme linje.

   Ved tokenisering blev nogle "ord" delt op, fx ".Svar." "." + "Svar" + "." . Andre "ord" blev derimod føjet sammen, fx "i form af" "i_form_af". Egentlig burde også følgende sammenføjninger ske: "m." + "fl." "m.fl.", "171." + "000" "171.000". Det var den anvendte tokeniser dog ikke i stand til.

   Disse opdelinger og sammenlægninger gjorde det meget svært senere at indsætte <Tn> og </Tn> markeringerne de rigtige steder: hvad der oprindeligt stod i teksten var der ikke mere, mens tekstens identitet netop var det eneste der kunne afgøre hvor markeringerne skulle sættes ind. Man kunne fx ikke bruge antallet af ordseparatorer (blanktegn) fra starten af sætningen til at bestemme markeringernes position.

   Problemet af kollationeringen (sammenfletningen) af søgeresultaterne med markeringer på den ene side og den taggede og parsede tekst på den anden, blev endnu større på grund af opdelingen af linjerne i sætninger, for dermed mistede vi også muligheden for at synkronisere på basis af linjenummer. Det betød at kollationeringen – når den røg af sporet – kunne have svært ved at komme på sporet igen.

5) Nogle specielle tegn optrådte ikke direkte i teksterne, men som en kode bestående af flere tegn, fx står "§" for " §". For at undgå at forvirre taggeren i dens morfologiske analyse er det generelt nødvendigt at foretage en konvertering fra koden til selve tegnet. Uheldigvis manglede det sidste tegn (semikolon) nogle gange i en kode når den stod i slutningen af en linje. Det skulle der tages højde for.

6) Nogle søgeresultater manglede af ikke kendte årsager markeringer. Dette gjaldt den første version af vid_data1.txt. I den anden version var problemet løst.

3.2.1 Konvertering til tokenliste

Kun linjerne med søgeresultater skulle analyseres, mens linjerne med søgeordene ikke skulle berøres. Derfor fik linjerne med søgeresultater en behandling som gjorde dem egnede til det næste trin i processeringen: tallet i starten af linjen og markeringerne i resultatteksten blev fjernet, og koder som udtrykker specielle tegn blev konverteret til disse tegn.

Den resulterende tekstfil blev overført til CST's tokeniser. Tokeniseren tog sig også af flerordsenheder. Flerordsenheder som "i alt" opfatter tokeniseren som ét token og binder det sammen med strege: "i_alt". Programmet tager også højde for punktummer i forkortelser. Andre punktummer opfattes som sætningsafgrænsere. Resutatet af denne proces er en fil med ét token pr. linje.

Efter tokenisering omdannede et andet program tokenfilen til en fil med én sætning pr. linje.

3.2.2 Ordklassebestemmelse (Brill)

Brill's tagger, med små modifikationer for bedre at håndtere ord som begynder med store bogstaver, tog førnævnte fil med én sætning per linje som input. Programmet var på forhånd blevet trænet til dansk og blevet kørt med disse parametre:

```
tagger FINAL.LEXICON.concat ..\sents.txt BIGBIGRAMLIST
  LEXRULEOUTFILE CONTEXT-RULEFILE
```

Denne tagger anvender Parole-tagsættet (se Bilag A).

3.2.3 Parsing (Cass)[6]

Den taggede tekst blev forberedt til input til Steve Abneys Cass-parser. Eventuelle forekomster af tegnene "[" og "]", som er reserverede tegn i parserens output, blev lavet om til "OPENSQUARE" og "CLOSESQUARE". I outputfilen fyldte hvert ord én linje. Linjens format er *ord* <tab> *POS-tag*. Sætningsafslutninger blev repræsenteret med en ekstra, tom, linje.

Denne fil blev overført fra pc til en Sun-workstation for der at afvikle parsningen, som anvender Cass-parseren med en grammatik som beskrevet i filen "np_pp_pp_gram.fsc" (se Bilag B). Outputtet af parseren er en i høj grad struktureret tekstfil, igen med højst ét ord per linje:

```
…

[ADV Herudover]
[V_PRES skal]
[NP
  [N kommunen]]
[V_INF etablere]
[NP2
  [NP1
```

[6] http://gross.sfs.nphil.uni-tuebingen.de:8080/release/cass.html

```
      [NP
        [PRON_UBST en]
        [N indsamlingsordning]]
      [PRÆP for]
      [NP
        [ADJ PVC-holdigt]
        [N affald]]]
    [PRÆP fra]
    [NP
      [N husholdninger]]]
  [ADV dog]
  [ADV ikke]
  [PRÆP fra]
  [TEGN .]

  …
```

3.2.4 Kollationering

Outputtet fra Cass blev kombineret med det oprindelige søgeresultat. Fx skulle førbeskrevne eksempel på Cass-output kombineres med dette søgeresultat:

```
… Herudover skal kommunen etablere en indsamlingsordning for PVC-holdigt <T9>affald</T9>
fra <T9>husholdninger</T9> dog ikke fra . …
```

Resultatet af denne kombination er:

```
[ADV Herudover]  [V_PRES skal]  [NP[N kommunen]]  [V_INF etablere]
[NP2[NP1[NP[PRON_UBST en]  [N indsamlingsordning]]  [PRÆP for]  [NP[ADJ PVC-holdigt]
<T9>[N affald]]] </T9> [PRÆP fra]  <T9>[NP[N husholdninger]]] </T9> [ADV dog]  [ADV
ikke]  [PRÆP fra]  [TEGN .]
```

3.2.5 Generering af output

Vi var specielt interesserede i at vide hvilke konstituenter søgeordene befinder sig i. Især ville vi gerne hurtigt kunne se om de ligger i samme konstituent og om denne fælles konstituent er en NP. Dertil skrællede vi alle strukturinformationer væk som ikke indeholder de markerede ord og vi brugte farver for nemt at finde rundt:

wght	#	***<WORD>husholdningsaffald affald husholdning</WORD><COUNT>113</COUNT>***
39.0	1079172	Skema 1 . **[NP1 [NP [EGEN <T9>Affald</T9>]] [PRÆP fra] [NP [N <T9>husholdninger</T9>]]]** . Kommunens udgifter til affaldshåndtering . Skema 1 omhandler kommunens udgifter til håndtering af **[NP1 [NP [N <T9>affald</T9>]] [PRÆP fra] [NP [N <T9>husholdninger</T9>]]]** . **[NP1 [NP [EGEN <T9>Affald</T9>]] [PRÆP fra] [NP [N <T9>husholdninger</T9>]]]** Kommunens udgifter til affaldshåndtering Kr ekskl moms inkl affaldsafgift Driftsudgifter til indsam-ling og transport Udgifter til
…	…	…
19.0	1192843	samt private og offentlige institutioner til enten genanvendelse eller deponering .

		Herudover skal kommunen etablere **[NP2 [NP1 [NP [PRON_UBST en] [N indsamlingsordning]] [PRÆP for] [NP [ADJ PVC-holdigt] [N <T9>affald</T9>]]] [PRÆP fra] [NP [N <T9>husholdninger</T9>]]]** dog ikke fra . dagrenovation , til enten genanvendelse eller deponering . Miljøstyrelsen udsendte i forbindelse med revision af affaldsbekendtgørelsen et orienteringsbrev
…	…	…
9.0	1192833	bebyggelser , hvor der til stadighed er mere end 2000 husstande (Ref. 17) . Papir . Kommunen skal have **[NP1 [NP [V_PARTC_PAST etableret] [N indsamlingsordning]] [PRÆP for] [NP [N <T9>affald</T9>]]]** fra **[NP1 [NP [N <T9>husholdninger</T9>]] i form [PRÆP af] [NP [N dagblade]]]** , distriktsblade , uge- og månedsblade , herunder fag- og medlemsblade , adresseløse tryksager , adresserede forsendelser og telefonbøger fra
-11.0	685206	Ved »**<T9>affald</T9>**« forstås enhver form for levnedsmiddel- , **<T9>husholdnings</T9>**- og driftsaffald , som fremkommer ved skibets normale drift og som bortskaffes løbende eller periodevis . Definitionen omfatter enhver
-11.0	598736	**<T9>Affald</T9>**« omfatter i denne bekendtgørelse enhver form for levnedsmiddel- , **<T9>husholdnings</T9>**-, eller driftsaffald , som fremkommer ved skibets normale drift og som bortskaffes løbende eller periodevis . »**[NP [EGEN <T9>Affald</T9>]]**

Denne output er i html-format og kan læses i en browser. For at lette den manuelle analyse er alle søgeresultater vægtet med følgende algoritme:

$$vægt = 10(I - N - O) + \overline{T}$$

hvor

$I =$ antal <Tn> tags som ligger indenfor en eller anden konstituent.

Konstituenter er ord eller grupper af ord som parseren har genkendt på basis af den valgte grammatik. I dette tilfælde er alle fundne konstituenter NP'er.

$N =$ antal konstituenter (dvs. NP'er) som indeholder <Tn>-tags.

$O =$ antal <Tn> tags som ligger udenfor enhver konstituent (dvs. udenfor NP'erne).

$\overline{T} =$ gennemsnittet af T-værdierne i søgeresultatet.

Vægten er stillet op således at resultater med de højeste vægte er de bedste ifølge hypotesen at et resultat som samler flere T-taggede ord i én konstituent (NP) er bedre en et resultat som spreder de T-taggede ord over flere konstituenter eller lader dem havne udenfor konstituenter. Og selvfølgelig er resultater med høje T-værdier bedre end resultater med lave. Læg også mærke til at T-taggede ord som ligger udenfor en

konstituent trækker vægten ned. Man kunne også have valgt blot at se bort fra disse ord i beregningen af vægten.

I det viste eksempel er tallene i linje #1192843 (”samt private og offentlige …”) således $I = 2$, $N = 1$, $O = 0$ og $\overline{T} = (9+9)/2 = 9$. Dermed bliver $vægt = 10(2-1-0)+9 = 19$

4 Semiautomatisk produktion af termlister

4.1 Automatisk processering af lister

Til brug for termharmonisering og ontologiopbygning indenfor sagsbehandling af fagområderne patenter og varemærker i IPR-virksomheden Zacco, har det været relevant at foretage en semiautomatisk produktion af termlister uddraget fra Zaccos tekstmateriale. Vi har til dette arbejde valgt at afprøve en metode som blev udviklet i STO-projektet til udvidelse af fagsprogligt ordforråd (Jørgensen et al. 2003).

Outputtet fra tekstprocesseringen beskrevet i kapitel 3 danner udgangspunkt for termlisten; grundlaget for udvælgelsen af termer er altså den liste af ord der forekommer efter lemmatiseringen, men som ikke findes i STO's almene ordforråd. Denne liste kaldes *termkandidatlisten*.

Den frekvenssorterede termkandidatliste er blevet gennemgået manuelt ved CST og uegnede termkandidater markeret; dette kan fx være almensproglige ord der ved et tilfælde ikke er med i STO, fx *honorarændring* og *helium*. Samtidig er lemmatisering og ordklassebetegnelser for kandidaterne blevet kontrolleret og oplagte slåfejl og ortografiske fejl blevet rettet.

Generelt har vi slettet person- og stednavne som ikke synes specifikt interessante for domænet (fx landenavne). Andre former for proprier er blevet på listen: navne på fagrelevante organisationer og institutioner er blevet bibeholdt, fx. *Sø- og Handelsretten* og *Patent- og Varemærketidende*.

Til at automatisk uddrage flerordtermer og kollokationer har vi kigget på den *pointwise mutual information* (Church and Hanks, 1989) af taggede bi-, trigrammer. Som værktøj brugte vi CMU-Cambridge Statistical Language Tool (Clarkson and Rosenfeld, 1997). De analyserede n-grammer var taggede med en reduceret mængde af tags, således at kun ordklasseinformation var opmærket for hvert ord. Vi har ekskluderet n-grammer som indeholdt ordklasser som tal og adverbier og vi har især fokuseret på n-grammer med høj indbyrdes information som bestod af flere substantiver (EGEN eller N) eller af en substantiv, en præposition og en substantiv. Vi fandt nogle flerordstermer og identificerede navne på udenlandske og danske organisationer, firmaer, adresser og standarder. Eksempler af de automatisk uddragede data er de følgende:

```
Burkina/EGEN Faso/EGEN
Eurasian/ADJ patent/N office/N
information/N disclosure/N document/N
den/PRON_DEMO ikke-registrerede/ADJ design/N
EF/EGEN  design/N
Skånefrist/N for/PRÆP design/N
```

4.2 Supplering af termlister

Efter denne manuelle gennemgang hvor oplagte fejl er blevet rettet, er termlisterne blevet overdraget til termeksperterne hos Zacco. Disse har gennemgået listerne og slettet yderligere lemmaer som de ikke mente var fagtermer inden for deres domæne. Endvidere har de suppleret termlisterne med hjælp fra interne ordlister og opslagsværker (16 % af termerne). Parallelt med denne supplering af termkandidatlisterne, har man på CST set på om listerne yderligere burde suppleres med lemmaer der i første omgang var sorteret fra automatisk. Mange af termerne i Zacco's dokumenter, især dem fra det juridiske område, er kodede i STO. Derfor er disse automatisk blevet slettet i første omgang. Vi har derfor kigget på de ord som ikke var blevet udvalgt som termer i første omgang og har valgt nogle af disse som mulige termer. Udvælgelsen er sket ved bland andet automatisk at finde ord som indgår som dele i allerede fundne sammensatte termer (baglæns sortering af ord).

4.3 Evaluering

Vi har evalueret den automatisk producerede liste af termkandidater ved at beregne *precision* og *recall* i forhold til den endelige termliste produceret ved hjælp af termeksperterne. *Precision* indikerer den procentdel af de foreslåede termkandidater som termeksperterne har godkendt som termer, og er 71,14%. Fejl i de automatisk producerede lister skyldes delvis forkert opmærkning af ord især i de sætninger som indeholder mange engelske ord, delvis det faktum at nogle af termerne er kodede i STO, og derfor ikke bliver taget i betragtning som mulige termkandidater. *Recall* bliver beregnet som den procentdel af alle relevante termer i dokumenterne som er blevet fundet automatisk, og er 77,24%.

5 Identifikation af tekststumper

Zacco A/S ønsker at effektivisere produktionen af deres dokumenter, samt at forbedre dokumenternes kvalitet. Der antages at der indgår fælles tekststumper i mange af firmaets standard dokumenter og breve. Disse tekststumper skal findes, opmærkes, navngives og organiseres i en passende struktur, således at medarbejderne nemt kan finde dem, eventuelt ved hjælp af naturligt sprog. Formålet med strukturering af tekststumper er at skabe standard-dokumenter og -breve ud fra passende tekststumper, som nemt og effektivt kan vedligeholdes og ajourføres.

Det første trin i arbejdet med tekststumper er at definere hvad en tekststump er og, dernæst, identificere tekststumper og markere de dokumenter, hvor tekststumper optræder.

Da det ikke har været muligt at definere a priori hvad en tekststump er, har vi, i første omgang, besluttet at definere en tekststump som en sekvens af mere end seks tokens, hvor en token er et ord eller et tegn. Vi er specielt interesserede i tekststumper som forekommer mindst to gange i de kollektioner af dokumenter som vi har fået af Zacco. Vi har tilpasset og/eller udvidet eksisterende standard UNIX-programmel, samt små perl-programmer som genkender bigrams og trigrams. Vi har udviklet programmer der genkender n-grams (i første omgang 6-grams, 10-grams, 15-grams, 20-grams, 25-grams, 30-grams og 35-grams, men de kan hurtigt tilpasses til at håndtere andre typer n-grams) i de eksisterende tekstkollektioner. Programmerne finder n-grams i tekster, tæller deres frekvens og udskriver disse resultater i en fil. Man kan dernæst finde hvilke filer de forskellige n-grams optræder i ved hjælp af UNIX grep-kommandoer.

I UNIX er der særlige regler for hvilke tegn der kan anvendes i filnavne og, som beskrevet i afsnit 3.2.1, er disse regler ikke overholdt i filnavnene for Zaccos dokumenter. Inden vi kunne anvende grep-kommandoer har vi skiftet alle filnavne for patentdomænet således at de følger reglerne for UNIX-filnavne.

Vi har afprøvet metoden til at identificere tekststumper på patentteksterne og har fundet frem til tekststumper af størrelse 10 til 35 tokens. Via grep-kommandoer har vi identificeret de dokumenter i patentteksterne hvor tekststumper optræder. Konklusionen af vores første undersøgelse er at de største teksttumper som optræder i flere dokumenter svarer til selvstændige paragraffer i patentdokumenterne. Desuden optræder disse paragraffer i filer, hvis navne er relateret til hinanden. Dette indikerer at paragrafferne i patentdokumenter ofte svarer til selvstændige tekstenheder, som Zacco-medarbejderne har genkendt og opmærket. Mindre tekststumper gentages på tværs af dokumenterne, men eksperter fra Zacco bør vurdere om de kan betragtes som selvstændige tekstenheder.

Vi har afsluttet det første arbejde med tekststumpidentifikation. Fremtidige aktiviteter vil være forbundet med arbejdet med termlister. Der anvendes i enkelte tilfælde synonyme termer i de nuværende standarddokumenter. Alle synonyme termer bør genkendes, og man bør erstatte synonymer med den ene term i tekstkollektionerne inden tekststumperne genkendes, således at identifikationen af tekststumperne bliver mere

præcist. Firmaets terminologi skal også anvendes i navngivning og måske strukturering af tekststumperne.

6 Konklusion og fremtidigt arbejde

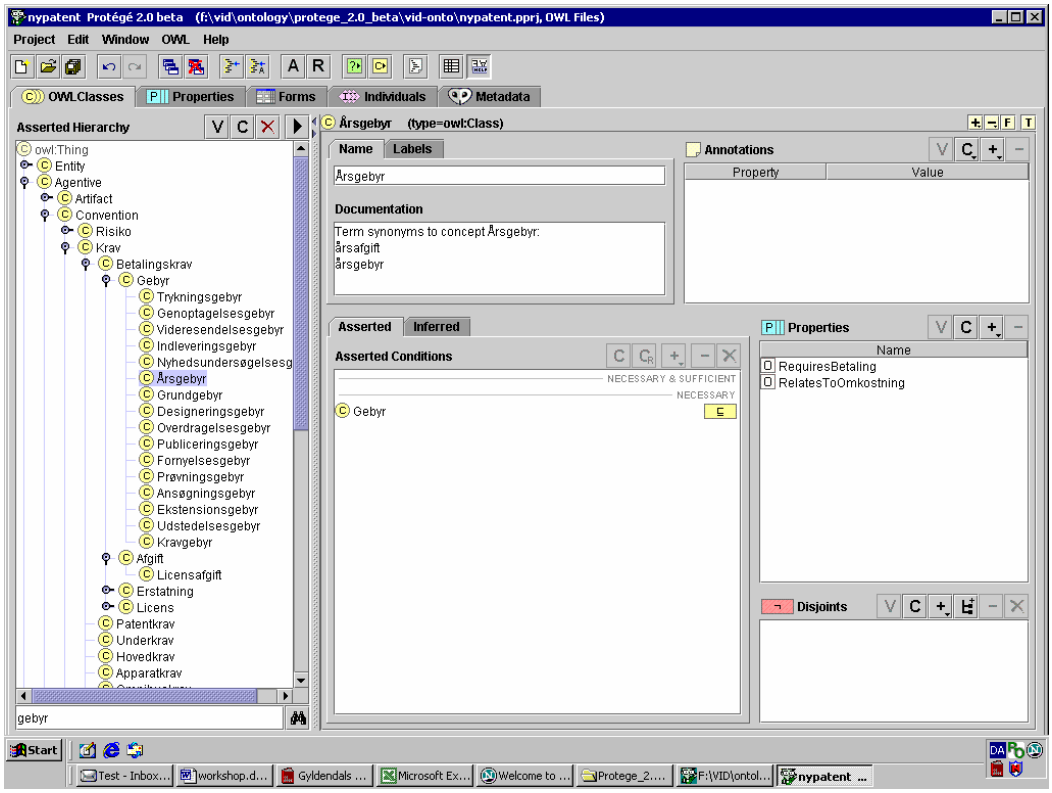
6.1 Evaluering af den automatiske analyse

Erfaringer fra den automatiske tekstanalyse har givet os den erkendelse at det ville have været fordelagtigt at tilpasse de sprogteknologiske værktøjer så de passede bedre til teksttypen; dvs. man kunne have udført en manuel vurdering af teksterne inden man påbegyndte det automatiske arbejde. Dette er især tilfældet for Zaccos tekster som har forskellige specifikke karakteristika i form af at de som nævnt er en slags halvfabrikata. Dette giver, som vi har set, problemer for værktøjerne som er konstrueret til 'almindelig' tekst.

En anden erfaring vedrører STOs egnethed som en slags stopordsliste for termidentifikation. STO indeholder en del såkaldte gråzoneord, og en del af disse ord – det gælder som nævnt især de juridiske termer – regnes for termer af termeksperterne og skal derfor ikke sorteres fra. Derfor er det tanken på længere sigt at raffinere metoden yderligere således at også eventuelle termordbøger inddrages i udtrækningsprocessen (jf. Jaquemin 2001). Ord der er i de relevante termordbøger og som forekommer i teksterne, bør altså bibeholdes på kandidatlisten – også selv om de er i STO.

6.2 Videre arbejde med Zaccos materiale

Hvad angår det fremtidige arbejde så vil den manuelle analyse af Zaccos data forløbe parallelt med den ontologiske analyse af domænevokabularet. Dette arbejde er påbegyndt, men endnu ikke afsluttet. Analysen vil ligesom produktion af termlisterne foregå i tæt samarbejde med termeksperterne. Nedenfor ses et udsnit af det forslag til en patentontologi der i første omgang blev opbygget ud fra patenttermlisten. Under dette arbejde afsløres bl.a. uhensigtsmæssige synonymer eller ujævnheder i terminologien, fx blev patenteksperterne via ontologiskitsen opmærksomme på at de to termer *årsafgift* og *årsgebyr* refererer til samme begreb (her etableret som Årsgebyr). Relationerne i ontologien etableres også ud fra tekstsammenhængen; i tilfældet med Årsgebyr er der etableret en relation til begreberne Betaling og Omkostning.



Parallelt med den manuelle ontologiopbygning er det også tanken at udforske clusteringmetoder til automatisk ontologiopnygning. Til dette arbejde er det imidlertid nødvendigt at opbygge større korpora, fx. på basis af patenttekster på nettet.

6.3 Videre arbejde med Ankiros materiale

Hvad angår Ankiros søgeudtræk med ord fra sammensatte udtryk, vil den manuelle analyse fokusere på følgende aspekter:

- holder hypotesen om at hits hvor begge søgeord optræder i samme NP generelt er gode hits ?
- hvad betyder afstanden ml. søgeordene inden for et NP mht. om et hit er godt eller dårligt ?
- hvilke syntaktiske og leksikalske principper holder for gode hits? fx:
- er nogle præpositioner 'bedre' end andre ?
- hvad betyder valens ? (valensbundne præpositioner vs. ikke valensbundne)
- typer af sammensatte ord: er deverbale sammesætninger mere variable end ikke deverbale sammensætninger ?
- ved deverbale sammensætninger: hvilke regler for disse.

Referencer

Church, K.W. and P.Hanks. 1989. Word association norms, mutual information and lexicography. In: *Proceedings of ACL 27*, pp.76-83.

Clarkson, P. and R. Rosenfeld. 1997. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proceedings of ESCA Eurospeech* 1997.

Haltrup D 2000. Træning og brug af Brill-taggeren på danske tekster. *ONTOQUERY Teknisk Rapport*, København (http://cst.dk/tagger/rapport/index.html).

Jacquemin, C. 2001. Spotting and Discovering Terms through National Language Processing. MIT Press. Cambridge, Massachusetss.

Jørgensen SW, Hansen C, Drost J, Haltrup D, Braasch A, Olsen S. 2003. Domain specific corpus building and lemma selection in a computational lexicon. *Proceedings of the Corpus Linguistics 2003 conference*, Lancaster, pp 374-383

Bilag A Parole tags

Dette er tagsættet som anvendes af POS-taggeren.

```
ADJ
ADJ_GEN
ADV
EGEN
EGEN_GEN
FORK
FORM
INTERJ
N
N_GEN
NUM
NUM_GEN
NUM_ORD
NUM_ORD_GEN
PRON_DEMO
PRON_DEMO_GEN
PRON_INTER_REL
PRON_INTER_REL_GEN
PRON_PERS
PRON_POSS
PRON_REC
PRON_REC_GEN
PRON_UBST
PRON_UBST_GEN
PRÆP
SKONJ
SYMBOL
TEGN
UKONJ
UL
UNIK
V_GERUND
V_IMP
V_INF
V_MED_INF
V_MED_PARTC_PAST
V_MED_PAST
V_MED_PRES
V_PARTC_PAST
V_PARTC_PRES
V_PAST
V_PRES
XX
```

Bilag B Cass-grammatik

Dette er np_pp_pp_gram.reg

```
:np
  determ = PRON_DEMO | PRON_UBST | PRON_INTER_REL ;
  attr = ADJ | FORK | UL | FORK SYMBOL | FORK ADJ | NUM | NUM_ORD |
V_PARTC_PAST | V_PARTC_PRES;
  gen = N_GEN | EGEN_GEN | ADJ_GEN | PRON_POSS |
           PRON_DEMO_GEN | PRON_UBST_GEN | NUM_GEN |PRON_REC_GEN |
PRON_INTER_REL_GEN;

  DP =  FORK? determ+ | determ SKONJ determ | gen SKONJ* gen* ;
  AP =  attr+|ADV? ADJ+ |ADV? attr TEGN ADV? attr|ADV? attr TEGN? ADV?
attr? SKONJ ADV? attr ADV? attr?;
  NP_G = DP AP* gen | DP* AP gen | EGEN gen;


  kerne = N | EGEN+ | NUM | UL | FORK| XX SKONJ N ;
  NP ->  ADJ? DP? AP? kerne | ADJ? NP_G? AP? kerne| PRON_DEMO attr |
NUM N N | PRON_UBST ADJ? N N | N EGEN+;


:np1
  PP = PRÆP NP;

  NP1 -> NP PP;


:np2
  PP = PRÆP NP;

  NP2 -> NP1 PP;
```