



UPPSALA
UNIVERSITET

Institutionen för lingvistik och filologi
Språkteknologiprogrammet
Examensarbete i datorlingvistik

3 juni 2005

En utvärdering av några konkordansverktyg

Marie Fryer Hydfors

Handledare:
Bengt Dahlqvist, Uppsala universitet

Sammandrag

I denna uppsats redogörs för ett antal konkordansprogram vilka är baserade på olika plattformar (DOS, Windows, MacIntosh, Linux samt Internet). Konkordansprogram används av språkteknologer, språkvetare, lärare, studenter och andra språkintresserade för att ta fram information om ett språk och dess användning. Några områden inom språkteknologin som kan ha användning av dessa program är bland annat: lexikografi, morfologi, kollokationsanalys, textanalys, vokabulärstudier samt korpusundersökningar. De elva program som är med i utvärderingen jämförs på ett empiriskt och kontrastivt sätt. Utvärderingen beskriver programmen på ett sådant sätt att de olika språkintresserade användare som finns ska veta vilket program som kommer att passa deras språkliga arbete bäst. De program som utvärderas i detta arbete är, med två undantag, ganska breda i sin utformning. Undersökningen visar att det kan vara mödan värt för den språkintresserade att byta plattform för att få tillgång till ett verktyg som hanterar fler funktioner. De verktyg som ska användas inom forskningen bör vara verktyg som kan hantera mycket stora korpusar.

Innehåll

Sammandrag	ii
Tack	vii
1 Inledning	1
1.1 Syfte	1
1.2 Disposition	1
2 Bakgrund	3
2.1 Korpusar	3
2.2 Konkordanser	3
2.3 Kollokationer	4
2.4 Vad är ett konkordansprogram?	5
2.5 Språkteknologiska användningsområden	5
2.5.1 Textanalys	5
2.5.2 Morfologiska studier	6
2.5.3 Kollokationsanalys	6
2.5.4 Vokabulärstudier	6
2.5.5 Lexikografiskt arbete	7
2.5.6 Korpusundersökningar	7
3 Metod	8
3.1 Utvärderingsmetod	8
3.2 Mål med utvärderingen	8
3.3 Utformning av utvärderingen	8
3.3.1 Urval av konkordansverktyg	8
3.3.2 Bedömningspunkter i undersökningen	8
3.3.3 Testmaterial	9
3.4 Genomförande av utvärdering	9
4 Konkordansprogrammen i undersökningen	11
4.1 Hum	11
4.1.1 Bakgrund	11
4.1.2 Specifik funktionalitet	11
4.1.3 Metoder och gränssnitt	12
4.1.4 Prestanda och begränsningar	12
4.1.5 Test på större texter	13
4.1.6 Resultat och omdömen	13
4.1.7 Användarstöd	13

4.2	Konk	14
4.2.1	Bakgrund	14
4.2.2	Specifik funktionalitet	14
4.2.3	Metoder och gränssnitt	14
4.2.4	Prestanda och begränsningar	14
4.2.5	Test på större texter	14
4.2.6	Resultat och omdömen	15
4.2.7	Användarstöd	15
4.3	TSSA	16
4.3.1	Bakgrund	16
4.3.2	Specifik funktionalitet	16
4.3.3	Metoder och gränssnitt	16
4.3.4	Prestanda och begränsningar	18
4.3.5	Test på större texter	19
4.3.6	Resultat och omdömen	19
4.3.7	Användarstöd	19
4.4	Textine	20
4.4.1	Bakgrund	20
4.4.2	Specifik funktionalitet	20
4.4.3	Metoder och gränssnitt	20
4.4.4	Prestanda och begränsningar	22
4.4.5	Test på större texter	22
4.4.6	Resultat och omdömen	23
4.4.7	Användarstöd	23
4.5	Conc för Mac	24
4.5.1	Bakgrund	24
4.5.2	Specifik funktionalitet	24
4.5.3	Metoder och gränssnitt	25
4.5.4	Prestanda och begränsningar	25
4.5.5	Test på större texter	25
4.5.6	Resultat och omdömen	25
4.5.7	Användarstöd	25
4.6	Monoconc	27
4.6.1	Bakgrund	27
4.6.2	Specifik funktionalitet	27
4.6.3	Metoder och gränssnitt	27
4.6.4	Prestanda och begränsningar	28
4.6.5	Test på större texter	28
4.6.6	Resultat och omdömen	28
4.6.7	Användarstöd	29
4.7	IMS Corpus Workbench	30
4.7.1	Bakgrund	30
4.7.2	Specifik funktionalitet	30
4.7.3	Metoder och gränssnitt	30
4.7.4	Prestanda och begränsningar	31
4.7.5	Test på större texter	31
4.7.6	Resultat och omdömen	31
4.7.7	Användarstöd	31
4.8	Kwic	32

4.8.1	Bakgrund	32
4.8.2	Specifik funktionalitet	32
4.8.3	Metoder och gränssnitt	32
4.8.4	Prestanda och begränsningar	33
4.8.5	Test på större texter	34
4.8.6	Resultat och omdömen	34
4.8.7	Användarstöd	35
4.9	Windex	36
4.9.1	Bakgrund	36
4.9.2	Specifik funktionalitet	36
4.9.3	Metoder och gränssnitt	36
4.9.4	Prestanda och begränsningar	37
4.9.5	Test på större texter	37
4.9.6	Resultat och omdömen	37
4.9.7	Användarstöd	37
4.10	Concordance	38
4.10.1	Bakgrund	38
4.10.2	Specifik funktionalitet	38
4.10.3	Metoder och gränssnitt	38
4.10.4	Prestanda och begränsningar	40
4.10.5	Test på större texter	41
4.10.6	Resultat och omdömen	41
4.10.7	Användarstöd	41
4.11	J-Bat Kwic	42
4.11.1	Bakgrund	42
4.11.2	Specifik funktionalitet	42
4.11.3	Metoder och gränssnitt	42
4.11.4	Prestanda och begränsningar	43
4.11.5	Test på större texter	43
4.11.6	Resultat och omdömen	43
4.11.7	Användarstöd	43
5	Resultatsammanställning	44
5.1	Klarar konkordansprogrammen testkorpusen?	44
5.2	Vilka plattformar kan programmen köras på?	44
5.3	Inom vilka språkteknologiska områden kommer programmen till nytta?	45
6	Diskussion	46
6.1	Konkordansprogrammets användbarhet	46
6.2	Vilka användare kan nyttja vilka program?	46
6.3	Skillnader i resultat	47
6.4	Sammanfattning	47
	Litteraturförteckning	49
A	Teckenrepresentationer	50
B	Länksamling	51
C	Bigramkörning i Textine	52

D Loggfil i TSSA	57
E Trigramkörning i TSSA	60

Tack

Först och främst vill jag tacka min handledare Bengt Dahlqvist för ämnet, likaväl som hans uppmuntran och stöd under mitt arbete med utvärderingen. Britt Marie Åhlenius och Jennie Gadeborg har hjälpt till att korrekturläsa materialet och det har varit mycket välkommet. På den mer tekniska sidan vill jag tacka Ola Engström för all hjälp med Macen, installation och goda råd. Daniel Fryer har också varit ett ypperligt bollplank och har fått svara på alla möjliga och omöjliga datoranknutna frågor från mig. Per Starbäck och ett flertal medstudenter (inga nämnda och inga glömda) har svarat på mina frågor i ”kommen” och i ”Chomsky”. Ett jättetack till er allihopa!

1 Inledning

Denna uppsats redogör för elva olika konkordansverktyg och hur de kan användas samt av vem och i vilket syfte. Inledningsvis kommer uppsatsen att kort beskriva de termer som ingår i merparten av programmen, därefter ges en analys av de olika konkordansverktygen. En och samma korpus, UNT-92, kommer att användas för flertalet¹ program och med hjälp av denna korpus kommer de olika programmens effektivitet att kunna bedömas. Flera exempel i form av körningar och deras resultat kommer att ges löpande i texten. Några utdrag ur fullständiga körningar kan den intresserade ta del av i bilagorna i denna uppsats. Avslutningsvis ges en grafisk sammanställning över konkordansverktygen och diskuterar där i vilket syfte applikationerna kan användas och vilka personer som kan tänkas ha nytta av dem.

1.1 Syfte

Syftet med arbetet är att utvärdera konkordansverktyg för att få reda på vilka verktyg som passar vilka användare. Olika konkordansverktyg har olika egenskaper, olika funktioner, och avsikten med denna uppsats är att redovisa de generella och specifika funktioner vilka innehas av verktygen i undersökningen. Konkordansverktyg är en givande källa till kunskap, mestadels språklig, men även ämnesmässig, för både språkteknologer, språkvetare, forskare, lärare och studenter. Några av verktygen är lättare att lära sig behärska än andra. Förhoppningen är att denna uppsats ska kunna leda till att den språkintresserade hittar det verktyg hon eller han behöver. Jag vill här poängtera att de verktyg jag har valt ut inte på något sätt representerar allt vad där finns i fråga om konkordansprogram.

1.2 Disposition

För att kunna tillgodogöra sig all information i denna uppsats och bättre förstå vad ett konkordansprogram kan innehålla inleds uppsatsen med en bakgrund (kapitel 2) och där redogörs för några centrala begrepp, nämligen korpusar, konkordanser och kollokationer. Därefter förklaras kortfattat hur ett konkordansprogram fungerar för att därpå beskriva några språkteknologiska områden där ett konkordansprogram kan vara användaren behjälplig. Metodavsnittet återfinns i det tredje kapitlet där jag redogör för hur utvärderingen har gått till. I huvudavsnittet (kapitel 4) går alla de program igenom som har utvärderats i undersökningen. De följer alla samma mall med avseende på de detaljer som analyserats. Kapitlet inleds med de aspekter som är generella

¹Samtliga program är inte möjliga att testa med denna korpus beroende på diverse tekniska begränsningar vilka återges i texten.

för alla konkordansprogram. I kapitel 5 görs en översiktlig resultat- sammanställning där läsaren snabbt kan göra sig en överblick över samtliga program. I det avslutande avsnittet (kapitel 6) diskuteras samtliga program och inom vilka språkteknologiska områden de kan vara tillämpliga. I Appendix återfinns en länksamling och teckenförklaringar. Bilagorna innefattar även utdrag ur några körningar.

2 Bakgrund

Konkordansprogram kan vara till en stor hjälp för studenter, forskare och andra språkintresserade då det handlar om att skapa möjligheter för en effektiv inläring eller att ta fram information om något visst lingvistiskt element. När ett språk finns dokumenterat i en korpus, det vill säga dokumenterat i mängder av texter, går det med hjälp av ett konkordansprogram att ta fram information om hur detta språk fungerar. Ett exempel på detta kan vara hur ord och fraser samverkar för att språket ska bli en helhet. En användare av konkordansprogram bör känna till begrepp såsom korpus, konkordans och kollokation för att kunna maximera sin inläring och få förståelse för programmet ifråga. Enligt (Kennedy, 1998) har datorernas intåg för att finna, sortera, analysera och kvantifiera lingvistiska särdrag och processer i enorma mängder data haft en stor betydelse för ett antal områden inom språkvetenskapen. Konkordansprogrammen finns i ett tämligen stort antal och de kan vara DOS-baserade eller anpassade för Linux- eller Windowsdatorer. Vissa program kan till och med utnyttjas på webben och är då oftast systemoberoende. Konkordansprogram används inom flera språkteknologiska och -vetenskapliga områden såsom textanalys, morfologiska studier, kollokationsanalys, vokabulärstudier, lexikografiskt arbete, korpusundersökningar och liknande.

2.1 Korpusar

En korpus är en mängd texter.¹ Dessa texter är ofta representativa för språket i stort och tillhör olika genrer (andra korpusar som existerar utan dessa egenskaper är till exempel så kallade spamkorpusar). De flesta korpusar har en ändlig mängd material medan en del är så kallade monitorkorpusar vilka byggs på efterhand. De flesta korpusar är också maskinläsbara, dock finns det undantag. En korpus kan vara annoterad, uppmärkt, på olika sätt, exempelvis lingvistiskt (ordklass, lemma, syntaktisk information etc) eller semantiskt (markering av ords släktskap respektive ords särdrag). Det finns också så kallade parallellkorpusar där två texter representerar samma innehåll på två skilda språk. En sådan parallellkorpus kan behandlas för att få länknings på stycke-, menings- och ordnivå, till exempel i ett konkordansprogram.

2.2 Konkordanser

En konkordans är ett utdrag ur en korpus. Konkordansen centreras kring ett nyckelord, omgivet av nyckelordets närmaste kontext. På engelska benämns detta nyckelord ofta KWIC, keyword-in-context. Enligt (Barnbrook, 1996) är KWIC-konkor-

¹Texterna består av skrivet eller nedskrivet talat språk.

danser inte det enda, men det mest förekommande förfaringssättet för att få fram ett utdrag ur en korpus. Om nyckelordet som söks i ett program återfinns på flera ställen i korpusen hamnar alla nyckelord under varandra, dock inte alltid totalt centrerade, i en lista. På detta sätt kan en användare genom en snabb överblick avgöra vilken ordklass som exempelvis förekommer flitigast efter ett visst verb. Här återges ett prov på en konkordans av ordet tänka:²

...a nu, käre lektorn. Men vi få [[tänka]] på att Elsa ska vara uringad...
...tre hem där. Lilla fröken ska [[tänka]] sig, att med en sån mor kan d...
...öken. Och då har man bara att [[tänka]] efter om man kan förtjäna någ...
...g och vin, men jag borde inte [[tänka]] på det just nu. Jag är bestäm...
...ningen. Det blir det. Och [[tänka]] sig att det ska vara Carl-Mag...
...kökspiga. Men det får en inte [[tänka]] på, när det nyttar pojken. Fö...
... skakar sin man. Jaså, kan [[tänka]]! Det är sånt du går och misst...
...djup, sjungande len klang. [[Tänka]] sig, kära! Hon säger! Hon säg...
...en med barnslig förvåning: [[Tänka]] sig! Det har hon vetat. Det h...
...ar, när han föddes. Kunde jag [[tänka]] på det? Han var ju så liten o...
...ändig på alla sätt. Kunde jag [[tänka]] på det? Skulle jag gå omkring...
...Markurell först nu kommit att [[tänka]] på att han inom kort skulle s...
...kslagen och förvirrad för att [[tänka]] på någonting annat än flykt. ...
...sa dem det. Jag har ingen att [[tänka]] på nu utan nu kan jag göra mi...

Här ses de fjorton förekomster som återfinns i korpusen. Av de fjorton förekomsterna av ordet "tänka" åtföljs nio av prepositioner och hela åtta av dessa prepositioner är ordet "på". Pronomen är också relativt vanligt; fyra förekomster av ordet "sig" återfinns i ovanstående utdrag. Den person som ska undersöka språket bör alltid utgå från en frekvenslista av en korpus och där välja ord som är intressanta för vidare undersökning. (Barnbrook, 1996) framhäver att ett konkordansprogram har två huvuduppgifter: att finna alla förekomster av nyckelord i en text samt att presentera resultatet av sökningen i ett passande format. Konkordanser är även en god grund för att gå vidare och undersöka kollokationer som observerats.

2.3 Kollokationer

En kollokation är enligt (Sinclair, 1991) förekomsten av två eller flera ord inom ett kortare avstånd från varandra i en text. Ytterligare definitioner på kollokationer återfinns i litteraturen och enligt (Smadja, 1993) är en kollokation en godtycklig och återkommande ordkombination. Dessa ordkombinationer är ofta domänberoende och återkommer i viss kontext. Enligt Smadja kan igenkänning av kollokationer vara svår då de ord som ingår i en sådan inte alltid återfinns bredvid varandra i en sats. Kollokationer återfinns i en konkordansrad där nyckelordet, ofta kallat nod, är i centrum. På var sida om noden räknas ett antal, ofta fem enligt Sinclair, ord som ingår i ett så kallat spann. Orden till vänster om noden numreras med ett minustecken framför och orden till höger numreras med ett plustecken. Här ses en konkordansrad ur (Barnbrook, 1996), vilket är ett (av mig nedkortat) lånat exempel:

²Monoconc användes till denna konkordans och korpusen är "Markurells i Wadköping". Nyckelorden är färgmarkerade i programmet och hamnar automatiskt inom hakparenteser vid en överföring till ordbehandlingsprogrammet Word.

towards	the	town,	as	a	place	where	I	could	most	easily
-5	-4	-3	-2	-1	nod	+1	+2	+3	+4	+5

Här kan vi se att noden är ordet "place" och vi ser också vilka ord som omger noden i detta spann. När användaren har fått fram dessa uppgifter tas de onummerade orden³ bort ur raderna. Därefter körs en frekvenslista på de kvarvarande orden i spannet för att se vilka ord som är mest frekvent förekommande tillsammans med "place". Högfrekventa ord såsom "the", "I", "and", "of" och så vidare är inte alltid intressanta att ta med och dessa ord kan i konkordansprogrammen oftast läggas in i en så kallad stopplista varefter dessa ord tas bort ur det slutgiltiga resultatet. Resultatet av en nods kollokat ses ofta i en lista där även olika statistiska mått ges (exempelvis förväntad spannfrekvens och observerad spannfrekvens). Problem som kan uppstå under arbetet med kollokationsanalyser är homografi, stavningsvariationer och lemmatisering. Dock är problemen inte olösliga, däremot kan de vara enormt tids- och resurskrävande från användarens sida.

(Kennedy, 1998) beskriver hur en av huvudkonsekvenserna för den korpusbaserade analysen resulterar i att gränsen mellan lexikalitet och grammatik suddas ut. Arbeten gjorda på kollokationer visar på nya definitioner av vad som utgör ett ord. Ibland är kollokationerna av en mer grammatisk art, ibland är den mer lexikal.

2.4 Vad är ett konkordansprogram?

Konkordanskörningar med hjälp av program är egentligen inte målet med språkteknologens eller den språkintresserades arbete, utan det är ett delmål på väg mot det slutgiltiga målet. Det stora målet är att undersöka ett visst språkligt fenomen. Resultaten som erhålls i konkordanser, frekvenslistor, kollokationer och så vidare används inom exempelvis parsning och taggning. De konkordansprogram som har undersökts i denna studie har mycket gemensamt med varandra. De flesta genererar olika typer av frekvenslistor, konkordanser och kollokationer. Se mer om detta under Konkordansprogrammen i utvärderingen (Kapitel 4). Dessutom har en del av programmen unika egenskaper som redogörs för under avsnittet Specifik funktionalitet under respektive program. Ytterligare en grund som skiljer programmen åt är på vilken plattform de kan köras. I undersökningen återfinns program för både Windows, DOS, Linux, MacIntosh samt program som kan nås på webben.

2.5 Språkteknologiska användningsområden

Det finns ett flertal olika språkteknologiska delområden där konkordansprogram kan komma väl till pass. Här nedan nämns några av dem och vad konkordansprogrammen kan göra för skillnad för respektive delområde.

2.5.1 Textanalys

Inom detta område studeras som namnet säger, texter av olika sorter. Studierna kan ske både manuellt och maskinellt och det är i det senare fallet som konkordanspro-

³Onummerade ord är de ord som förekommer i konkordansen, utanför spannet, det vill säga före -5 och efter +5.

gram kan komma till användning. Utgångspunkten i textanalysen är att ta fram nyckelord. Texter kan exempelvis analyseras med avseende på översättningar. Hur stor frihet har en översättare tagit sig när hon eller han tagit sig an ett verk att översätta? Texter kan också analyseras då ett verk moderniserats. Bibeln är ett exempel på en skrift som uppdaterats. Det svenska språket har förändrats mycket från 1917 till idag (den senaste kompletta utgåvan kom år 2000) och det märks naturligtvis på de skilda översättningarna.

I textanalysen ingår lexikalisk analys (där ordformer är det centrala), morfologisk analys (med ordformers egenskaper i fokus) samt syntaktisk analys (där fraser identifieras och deras egenskaper bestäms). Dessa nämnda områden är rent lingvistiska. Textanalysen vänder sig också till semantiskt intresserade i och med möjligheten att göra innehållsanalyser. Då behandlas istället innehållet i den mänskliga kommunikationen, det vill säga, vad det är som författarna ifråga vill överföra för budskap till sina läsare.

2.5.2 Morfologiska studier

Ett morfem är språkets minsta betydelsebärande enhet. Inom ämnet morfologi studeras skilda morfemtyper (det kan vara segmentella, prosodiska, osynliga eller tomma morfem, samt kombinationer av dessa). Morfologer är också intresserade av att studera skillnader mellan böjningar och avledningar, sammansättningar, ords produktivitet samt gränssnitt mellan morfologi och andra läror (prosodi, syntax etc). Med hjälp av ett konkordansprogram kan morfologen ta fram mängder av fakta av lexikografisk natur att analysera.

2.5.3 Kollokationsanalys

Att analysera kollokationer (två eller flera ord som förekommer inom ett kortare avstånd från varandra) är ett lexikografiskt arbete som för att bli tillfredsställande kräver en dator och en korpus. Språkforskaren utgår ofta från ett förväntat resultat och kan efter sitt utförda arbete jämföra detta med det faktiska innehållet. Mer om kollokationer finns att läsa i sektion 2.3.

2.5.4 Vokabulärstudier

Vokabulärstudier handlar i stort om att lära sig hur ett nytt ordförråd lärs in. Ofta ligger då tyngdpunkten på hur ett andraspråk lärs in. I Wales finns en forskningsgrupp, Cals, som är högt specialiserad på vokabulärstudier.⁴ Cals (Centre for Applied Language Studies) har flera intressanta länkar, bland annat till Varga (Vocabulary Acquisition Research Group Archive), där olika typer av vokabulärstudier finns representerade. Ämnen som avhandlas där är exempelvis flerspråkiga lexikon, förhållandet mellan ordförråd och läsförståelse på andraspråket samt metaformedvetenhet och bibehållande av vokabulären. Vokabulärstudier sammanfaller delvis med lexikografiskt arbete.

⁴<http://www.swan.ac.uk/cals/calsres/index/> - här finns exempelvis ett vokabulärtest på engelska för den nyfikne att prova.

2.5.5 Lexikografiskt arbete

Lexikografer arbetar med att framställa lexikon av skilda slag. De tar reda på hur vanligt förekommande vissa ord är, hur vanliga olika betydelser för ett givet ord är, om ord har systematiska associationer med andra ord och om ord har systematiska associationer med speciella register⁵ eller dialekter. Inom den datorlingvistiska grenen sysslar lexikografer med att utvinna information ur korpusar, så kallad text data mining, automatisk maskininlärning från texter (med semantisk och syntaktisk information), återanvändning av maskinläsbara lexikon och datorstödd framtagning av ordböcker. Lexikografer arbetar också med framställan av så kallade ordnät, till exempel Svenskt OrdNät⁶ (en del av WordNet) vilken beskriver ords semantiska relationer.

2.5.6 Korpusundersökningar

I korpusbaserade undersökningar kan forskare ha flera delmål eller mål. De kan se rent lingvistiska aspekter, såsom hur språket förändras över tid, eller innehållsaspekter på ett språk. Jurister, ekonomer eller andra yrkesgrupper med lingvistiskt intresse kan tänkas använda sig av korpusundersökningar då de letar efter fenomen i sina domäner och hur de förändras över tid. Ett exempel på detta kan vara hur vanligt det är att media skriver om en viss företeelse, exempelvis gängbrottslighet. De kan då vända sig till en större tidning och be att få ut samlade korpusar för att jämföra hur ofta förekommande ordet gängbrottslighet var med avseende på åren 1990 och 2000. Först ställer sig säkerligen juristen hypotesen att ordet förekommer flitigare i den senare korpusen. Därefter gör hon eller han en korpusundersökning som verifierar alternativt kullkastar den teorin. Korpusundersökningar blir allt vanligare (Biber, 1998) och alltmer lättillgängliga för intresserade användare. Biber tar också upp andra områden där korpusundersökningar förekommer och nämner bland annat lexikografi och grammatik som två områden där korpusar är av stor betydelse för forskningen. Däremot anser inte Biber att den korpusbaserade undersökningsmetoden ska vara den allenarådande metoden utan ses som ett komplement till mer traditionella synsätt. Som ett exempel nämner Biber att det är möjligt att utgå från teorier, exempelvis att vissa lingvistiska särdrag anammas av barn vid en viss ålder. Dessa teorier kan prövas i en därför avpassad korpus.

⁵Ett register kan vara människors mål, mening, sätt eller interaktivitet med språket. En människa varierar mellan dessa register under dagens lopp.

⁶En länk (äldre) till projektets hemsida är: <http://www.ling.lu.se/projects/SWordNet/>

3 Metod

3.1 Utvärderingsmetod

Jag har valt att redovisa ett antal konkordansverktyg på ett empiriskt och kontrastivt sätt. Detta för att så småningom komma fram till vilka verktyg som passar olika användare bäst. Tyngdpunkten av undersökningen ligger på programmens funktionalitet.

3.2 Mål med utvärderingen

Målet med denna utvärdering av dessa program är att ta fram ett kunskapsunderlag för språkteknologer, språkvetare och andra språkintresserade då de behöver ett konkordansprogram i sitt arbete eller till sin utbildning. Detta kunskapsunderlag består i relativt utförliga beskrivningar av programmen i undersökningen samt några i slutet medtagna figurer i vilka en snabbsökning av information är möjlig. I dessa figurer kan användaren jämföra vilket program som bäst passar för hennes eller hans rådande arbetsuppgift.

3.3 Utformning av utvärderingen

3.3.1 Urval av konkordansverktyg

De program som är medtagna här representerar ett brett urval av program anpassade för olika plattformar: DOS, Windows, MacIntosh, Linux och Internet. Målet med detta plattformsurval var att få med så många skilda typer av konkordansprogram som möjligt. Eftersom användare av konkordansprogram har tillgång till olika typer av operativsystem var det nödvändigt att fånga upp program som fungerar på olika plattformar. Det ska också sägas att detta urval av konkordansprogram ändå är en liten del av alla förekommande program. Nya och påbyggda program kommer ständigt ut på marknaden, både som gratisprogram och fullversioner som användaren måste betala för. Många av de verktyg som har undersökts är verktyg som finns åtkomliga på Institutionen för Lingvistik och filologi och ett fåtal av dessa verktyg är av olika skäl inte de senast utkomna.

3.3.2 Bedömningspunkter i undersökningen

De punkter som går igenom för varje program i denna undersökning är:

- *Bakgrund* - Kort avsnitt om vem som skapat konkordansprogrammet, vid vilken tidpunkt samt vad programmet gör.
- *Specifik funktionalitet* - Detta avsnitt tar upp vad som särskiljer detta program från andra i samma genre. Det kan vara rent lingvistiska aspekter eller skillnader i programmens gränssnitt.
- *Metoder och gränssnitt* - Här beskrivs relativt utförligt vad programmet innehåller för delar och hur de används. Denna del ska inte ses som en uttömmande beskrivning av programmet ifråga.
- *Prestanda och begränsningar* - Avsnittet behandlar programmets plattform, storlek (i den mån detta går att utröna via programmet) samt vad programmet klarar av att utföra.
- *Test på större texter* - I denna del jämförs en och samma korpus på samtliga¹ i undersökningen ingående program.
- *Resultat och omdömen* - Här noterar jag mina erfarenheter av programmet. Denna del ska ses som författarens, det vill säga min, subjektiva åsikt.
- *Användarstöd* - Till nästan alla program hör någon form av manual. Denna punkt beskriver kort hur den är beskaffad.

3.3.3 Testmaterial

De mindre korpusar som använts i arbetet då exemplifieringar görs är ”Markurells i Wadköping”² och ”Hamlet”³. I det fall program för MacIntosh (Conc för Mac, sektion 4.5) har granskats har korpusen varit det första kapitlet i medföljande ”Alice in Wonderland”⁴. Det nätbaserade programmet J-bat Kwic (sektion 4.11) nås via en hemsida där man initialt måste välja en korpus för att kunna prova konkordansprogrammet över huvud taget. Dessa givna korpusar består av engelskspråkiga tidningar samt en del skönlitteratur. IMS Corpus Workbench (sektion 4.7) har testats med SUC-korpusen⁵. Den slutgiltiga testningen som utförts på programmen i undersökningen har utgått ifrån UNT-92⁶, en korpus med samtliga texter som skrivits i Upsala Nya Tidning under 1992.

3.4 Genomförande av utvärdering

Jag har gått igenom samtliga program och gjort diverse empiriska körningar för att utvärdera programmet och bilda mig en uppfattning om hur de är konstruerade och vad de klarar av. När samtliga program var testade och beskrivna använde jag mig av

¹Vissa undantag måste göras på grund av tekniska begränsningar.

²Verket är skrivet av Hjalmar Bergman 1919, 386 kB. Antal ordformer i korpusen är 64.188 stycken.

³Verket är skrivet av William Shakespeare 1602, 176 kB. Antal ordformer i korpusen är 32.034 stycken.

⁴Verket är skrivet av Lewis Carroll (pseud) 1865. Antal ordformer i korpusen är 2.172 stycken.

⁵SUC står för ”Stockholm Umeå Corpus” (Ejerhed, Källgren, Wennerstedt och Åström, 1992).

⁶UNT-92 är 41.3 MB stor och innehåller ca 6,5 miljoner ordformer.

samma korpus, UNT-92, för att se hur programmen klarade av att utföra vissa uppgifter då det handlade om en klart större datamängd. Det är dessa uppgifter som sedan sammanställts i diskussionen (kapitel 6) för att möjliggöra för presumtiva användare vilka program som kan vara användbara i deras språkteknologiska och språkvetenskapliga arbetsuppgifter.

4 Konkordansprogrammen i undersökningen

De program som är medtagna i denna undersökning körs på olika plattformar: på Internet, i DOS, i Windows, i Linux eller på MacIntosh.

I undersökningen ingår följande konkordansprogram:

- Hum
- Konk
- TSSA
- Textine
- Conc för Mac
- Monoconc
- IMS Corpus Workbench
- Kwic
- Windex
- Concordance
- J-bat Kwic

Programmen har utvärderats empiriskt. Samtliga funktioner i programmet har testats och användarstödet har gått genom och fått en kort kommentar.

4.1 Hum

4.1.1 Bakgrund

Hum är flera små program, utvecklade i Unix-miljö och skapade av Bill Tuthill under 1980-talets allra första år. Den version som testats här, i ett Linux-system, är från 1992 och är avsedd för att användas i språklig databehandling, speciellt för att skapa konkordanser och stöddokument.

4.1.2 Specifik funktionalitet

Med Hum är det möjligt att vända på konkordanser samt att se på ord-, tecken- och digraffrekvenser (alltså frekvenser över två tecken i taget).

4.1.3 Metoder och gränssnitt

Hum består som ovan nämnts av flera små program, närmare bestämt 25 stycken (se figur nedan). Programmen startas var för sig och till detta används ett terminalfönster. I manualen kan användaren läsa sig till hur programmen används och vilka förändringar man kan göra i de olika programmen. För att användaren ska kunna optimera sitt arbete är det möjligt att göra flera typer av inställningar med hjälp av så kallade flaggor med valfria argument. De metoder som finns att tillgå räknas upp här (så här ser det ut på skärmen då användaren skriver ”man-hum” efter prompten):

hum-accent	hum-freq	humount	hum-skel	hum-troffmt
hum-cedilla	hum-kwal	hum-pair	hum-togrkr	hum-umlaut
hum-dict	hum-kwic	hum-pause	hum-tolpr	hum-wdlen
hum-exclude	hum-lno	hum-revconc	hum-tosel	hum-wheel
hum-format	hum-maxwd	hum-sfind	hum-tprep	hum-xref

De metoder som har undersökts närmare är:

- hum-freq file - ger en frekvensordlista av en text
- hum-kwic file - ger en konkordans av en text
- hum-kwal file - ger en nyckelords- och konkordanslista av en text
- hum-revconc file - ger en omvänd konkordans

Frekvensfilen listar orden efter förekomst och ger svar på hur många ord som finns i filen. Dessutom anges hur många olika ord korpusen innehåller. Om användaren vill räkna ut förhållandet mellan ordformer och ordtyper får detta ske manuellt.

Nyckelords- och konkordanslistorna är inte centrerade. De är något svårslästa i sitt normalutförande, men med angivande av flaggor är det möjligt att förändra resultatets utseende.

Den omvända konkordansfilen är ingen egentlig omvänd konkordansrad. Om användaren skriver in ”varför är bladen gröna” i standard input, blir resultatet ”röfrav är bladen gröna”. Det är alltså det inledande ordet på varje rad som blir omvänt.

Det går bra att skapa egna stoppfiler, liksom inkluderingsfiler. Programmet kan även radnumrera en korpus såväl som att lokalisera, mäta och skriva ut korpusens längsta ord. Två filer kan också parallellköras i programmet hum-pair. Det finns ett Unix-verktyg som heter grep, med vilket man söker efter specifika strängar. I Hum finns en motsvarighet där användaren kan söka efter ett speciellt mönster med hjälp av hum-sfind. Denna sökning opererar dock på meningar snarare än rader.

4.1.4 Prestanda och begränsningar

Hum, som är körbart på Linux-system, saknar räknare för sina operationer. Dock är programmet utvecklat på så sätt att det går att använda flera typer av alfabetet. Programmen hum-accent, hum-umlaut och hum-cedilla är behjälpliga då andra tecken än de ingående i det engelska alfabetet används. Hum läser såväl från filer som från standard input, dock saknas en prompt att skriva in sin text efter. Markören hoppar endast över till en ny rad i terminalfönstret där användaren förväntas skriva in sin text.

4.1.5 Test på större texter

UNT-92 har provats framgångsrikt på denna samling program. Det tog en knapp minut för hum-freq att ladda in och ordna korpusen efter frekvens.

4.1.6 Resultat och omdömen

Hum får sägas vara avancerat med tanke på tidpunkten då det är skapat, men programmen uppfattas inte som lätta att använda. Brukaren av programmen måste använda sig av terminalfönstret i ett Linux-system för att köra Hum. För att kunna utnyttja programmen maximalt behöver användaren vara kunnig i hur textmassor ska sorteras, sparas, hämtas och lagras. Konkordanserna blir inte centrerade på skärmen, vilket kan göra resultatet svårsläst. Det är omständligt att behöva skriva in så många flaggor, inklusive argument, för att kunna styra sin sökning. Som ett exempel kan nämnas att i den del av Hum där kwic skapas (hum-kwic), finns tolv olika flaggor att tillgå.

4.1.7 Användarstöd

Varje delprogram har en manual där användaren får utförlig information om vad programmet innehåller, vilka flaggor och argument som ska användas, vad programmet är till för samt ofta också vilka buggar som kan tänkas uppstå under körning. Varje liten manual har också en referens till närliggande program ("see also"), men ett flertal stämmer inte då det saknas manualer för dem.

4.2 Konk

4.2.1 Bakgrund

Programmet är framtaget av Per Starbäck vid institutionen för lingvistik och filologi vid Uppsala universitet. Konk är anpassat för arbete med enbart konkordanser. Skriptet är delvis anpassat för en student vid nämnda institution för arbetet med en D-uppsats. Versionen som har testats här är från 19 maj 2004.

4.2.2 Specifik funktionalitet

Detta program kan hantera taggade korpusar.

4.2.3 Metoder och gränssnitt

Konk startas via ett terminalfönster på en Linuxdator och användaren skriver in kommandorader. Dessa rader kan innehålla lämpliga flaggor, beroende på vad användaren vill få fram för information. Resultatet (utfilen), liksom infilen, ska stå med i kommandoraden. Resultatet av en körning kan ses i en editor, exempelvis i ett emacs-fönster. Det är viktigt att spara om filerna under olika namn, annars skrivs äldre körningar över automatiskt. De typer av konkordanssökningar som användaren kan få fram med hjälp av dessa flaggor är:

- sorteringsordning för konkordansraderna
- bestämning av radlängder
- bestämning av längder på mittkolumnen
- indikation huruvida korpusen är taggad (reguljära uttryck används)
- gömmande av taggar vid utskrift

4.2.4 Prestanda och begränsningar

Konk är anpassat för Linux-system och är relativt litet till storlek och användningsområde (enbart konkordanser). Programmet saknar räknare vilket gör att det inte går att beräkna tidsåtgång för körningar. Det finns inte heller någon typ av summering för hur många träffar som återfinns vid en konkordanssökning. Nyckelorden centreras på raderna i editorn vilket gör sökningen lättläst.

4.2.5 Test på större texter

Konk behövde en knapp minut för att ladda in UNT-92-korpusen och sökningen lyckades utan problem.

4.2.6 Resultat och omdömen

För en person som är van vid Linux-system kan detta program vara nyttigt och då speciellt om korpusen som används är taggad. Konk är ett program som är enkelt men förmodligen utbyggbart. Ofta behöver en användare veta hur många träffar en viss sökning genererade.

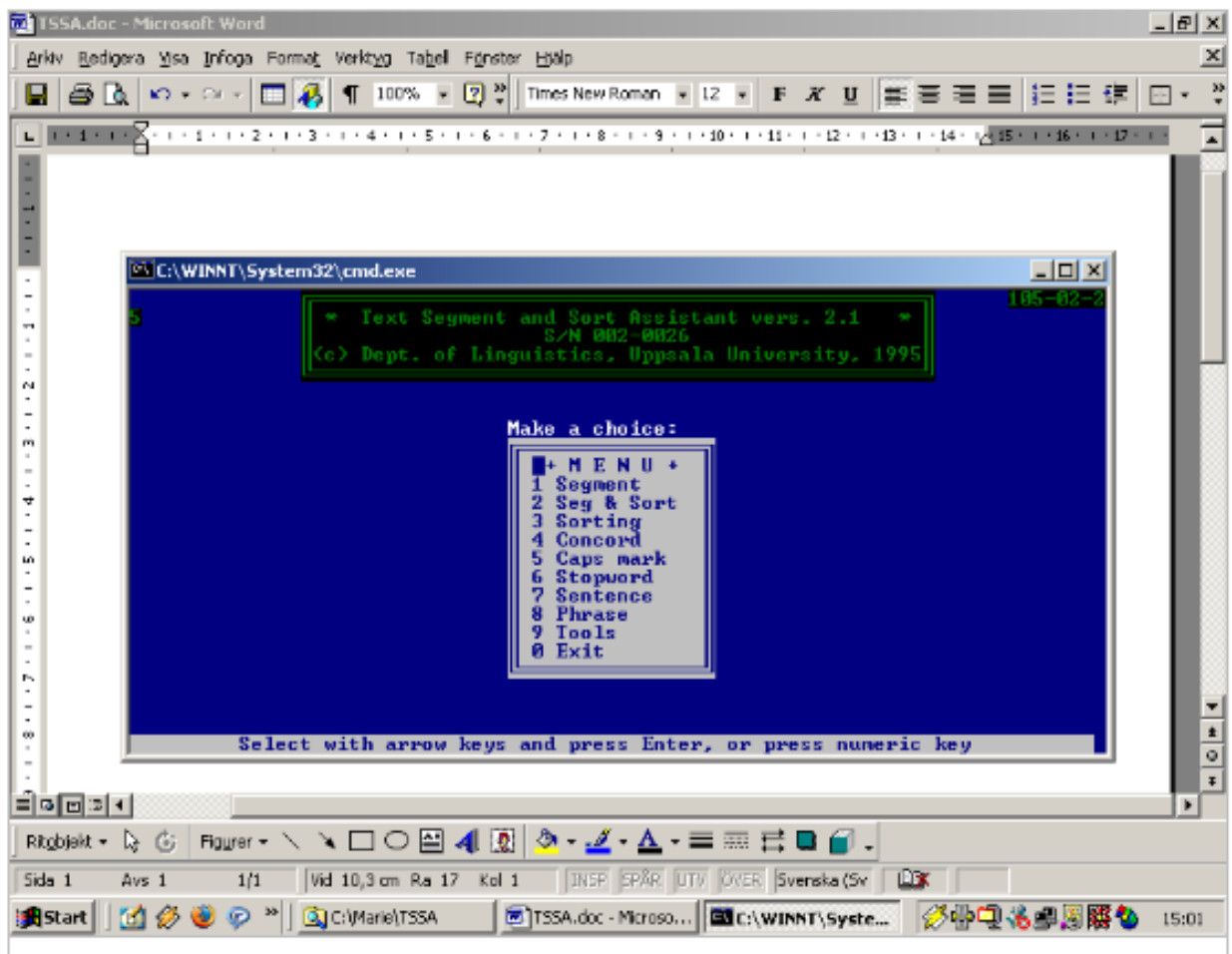
4.2.7 Användarstöd

Användarstödet består av en enda sida som kan tas fram i terminalfönstret via kommandoraden:

```
konk --help
```

Det står skrivet att infilen ska skrivas i standard input vilket inte är korrekt. För att framgångsrikt kunna köra konk behövs en korpus som infil. Eftersom konk inte är tänkt att utnyttjas utanför institutionen, så kanske dessa fel inte är så allvarliga.

4.3 TSSA



4.3.1 Bakgrund

TSSA skapades 1994 och är ett DOS-program avsett för textsegmentering och sortering. Upphovsman till detta program är Bengt Dahlqvist vid Institutionen för lingvistik och filologi vid Uppsala universitet. TSSA bygger på ett menysystem där användaren kan skapa bland annat ordlistor och konkordanslistor efter önskade sorteringskriterier. Dessutom är det möjligt att generera flerspråkiga meningslänkar, ta fram fraser och även hantera uppmärkning.

4.3.2 Specifik funktionalitet

TSSAs unika egenskap är att programmet klarar av att generera flerspråkiga meningslänkar. Då detta inte är ett av undersökningskriterierna så måste det förbli osagt hur effektivt TSSA är i detta avseende.

4.3.3 Metoder och gränssnitt

TSSA (en akronym för Text Segment and Sort Assistant) är ett program som körs i ett DOS-fönster. Användaren brukar piltangenterna på tangentbordet för att manö-

vrera sig i menysystemen. I fönstret kan användaren välja mellan olika uppgifter i huvudmenyn. Vad som går att välja mellan beskrivs nedan.

1) Segment

För att kunna segmentera text behöver programmet indata från en korpus. Med hjälp av piltangenterna väljs en lämplig korpus. Denna måste finnas i samma katalog som själva programmet. När namnet på utfilen angetts får denna automatiskt filändelsen . wrd. Utfilen består av hela korpusen utskrivna med ett ord på varje rad. Programmet innehåller de skiljetecken som är default (.,:;?!'()/). Programmet talar därefter om hur många ord som finns i korpusen och frågar vidare om användaren vill skapa en tvåkolumnslista av samma fil. Denna lista ges ändelsen .lst. Här kan användaren tala om hur många ord som ska stå på varje sida. En .log-fil skapas samtidigt som . wrd-filen. Loggfilen innehåller information om antal ord i texten, antal tecken i texten (inklusive blanksteg), antal definierade skiljetecken, antal fördefinierade tecken, längdfördelning på ord i texten (i frekvens och procent), ordmedellängd (inklusive avvikelser och räckvidd). Det finns också en frekvensfördelning för alla tecken, antal tecken i texten samt även frekvensfördelning för okända tecken och antal odefinierade tecken. Ett exempel på en loggfil kan ses i bilaga D.

2) Seg & Sort

Under denna rubrik kommer ytterligare en meny upp. Här anger användaren vilken typ av sortering som ska göras: "Initial", "Final", "Final w/i", "Frequency" eller "Occurrence".

"Initial" innebär att alla ord som skrivs ut i filen är alfabetiskt ordnade med de ord som börjar på a överst i filen. Ändelsen på denna fil är .ini.

När man jobbar med "Final" så skrivs orden ut i alfabetisk ordning med alla ord som slutar på a först. Ändelsen på denna fil är .fin

"Final w/i" innebär samma sak som "Final" med det tillägget att orden är högerjusterade. Även dessa filtyper får ändelsen .fin.

"Frekvens" ger en frekvenssorterad lista med ändelsen .fre. Här kan användaren dessutom välja nedre gräns för frekvens.

"Occurrence" listar orden i den ordning de förekommer i korpusen. Orden tas endast med i listan en gång. Samtliga undermenyer i "Seg & Sort" genererar loggfiler där statistik över frekvens och annan information kan utläsas.

3) Sort

Denna rubrik har en undermeny som är likadan som "Seg & Sort". För att kunna arbeta vidare i undermenyn måste en . wrd-fil finnas tillgänglig i katalogen. Detta gäller för samtliga val i "Sort". Därefter kan samma uppgifter som i "Seg & Sort" utföras, det vill säga: "Initial", "Final", "Final w/i", "Frequency" eller "Occurrence".

4) Concord

I denna meny finns fem undermenyer att välja på. Det är möjligt att skapa olika typer av konkordanser från korpusen. Användaren kan ordna konkordanser efter förekomst,

ordna alfabetiskt efter den efterföljande kontexten, förkorta en konkordanslista efter en inkluderad eller exkluderad .con-fil. De filer som skapas får extensionerna .con. Det är också möjligt att skapa konkordanser med hela meningar från en ordlista. Här krävs en .sen-fil som input samt en .rdc-fil, det vill säga en meningsnumrerad konkordanslista. Detta blir en blank fil eftersom DOS inte beter sig på samma sätt under Windows 2000 och Windows XP som den gjorde i det system där TSSA skapades. Kompilatorn finns inte längre tillgänglig varför detta problem inte kan lösas.

5) Caps mark

Användaren väljer denna rubrik när korpusen behöver gås igenom och eventuella versaler görs om till gemener. Här är det möjligt att gå igenom en .txt-fil (direkt i programfönstret), det vill säga en korpus och markera versaler. Hela texten måste läsas igenom, det går inte att påbörja läsning på valfri plats i korpusen.

6) Stopword

Detta val kan göras då man vill förkorta en redan existerande ordlista, en .ini-fil. Till detta behövs en .stp-fil, exempelvis en redigerad frekvenslista. Resultatet av detta arbete blir .out-filer.

7) Sentence

Till denna rubrik hör en undermeny där ett flertal val är möjliga. Användaren kan välja mellan att segmentera (korpusfilen), dela (.sen-fil), sammanfoga (.sen-fil), sammanfoga en mängd (.sen-fil och .bjn-fil) samt slå samman två filer (två .sen-filer). Det centrala är att segmentera en textfil i meningar och denna funktion är avsedd för flerspråkiga alignments av filer.

8) Phrase

I denna meny finns två val: att hitta respektive markera fraser. Användaren anger hur många ord som ska förekomma i frasen (defaultvärdet är 2). Användaren får även ange lägsta frekvensgräns (default är här 1). Input här är en korpus och output är en frasfrekvensfil (.phr).

9) Tools I kategorin Tools kan användaren ställa in personliga önskemål för hur programmet ska kunna användas. För mer information om detta, se (Dahlqvist, 1997).

4.3.4 Prestanda och begränsningar

TSSA upptar 756 kB på hårddisken. Applikationen är snabb för en användare som inte begagnar sig av alltför stora texter. Ofta står det hur lång tid en viss process har tagit (mer om detta under "Test på större texter"). Begränsningarna i systemet är få och härrör sig då till DOS-miljön.

Det alfabet som är möjligt att använda är det latinska. TSSA använder inte Windows ANSI-kodning så användaren får själv gå in manuellt i .inf-filer och där se till att önskade tecken står med (dessa kan sorteras efter eget önskemål). De infiler som är möjliga talar TSSA själv om via fönstret. Inledningsvis är det alltid fråga om en korpus i .txt-format. Utfilerna blir oftast, men inte alltid, tilldelade ändelser automatiskt.

4.3.5 Test på större texter

TSSA kan hantera en så stor fil som UNT-92, men det tar mycket lång tid att få körningarna utförda. Att läsa in själva filen tog 13 minuter och då jag därefter valde att sortera initialalfabetiskt tog det drygt 67 minuter att få fram ett resultat. TSSA visade att det finns 323.456 ordtyper i filen och 6.570.913 ordformer sammanlagt. Förhållandet (ratio) häremellan är alltså 0.049, enligt TSSA. Jag gick vidare och sökte efter samtliga fraser som innehåller tre ordformer. Programmet tog oerhört lång tid på sig, drygt 219 minuter, men ett resultat på sammanlagt 129MB gick att frambringa (se bilaga E för delar av denna körning).

4.3.6 Resultat och omdömen

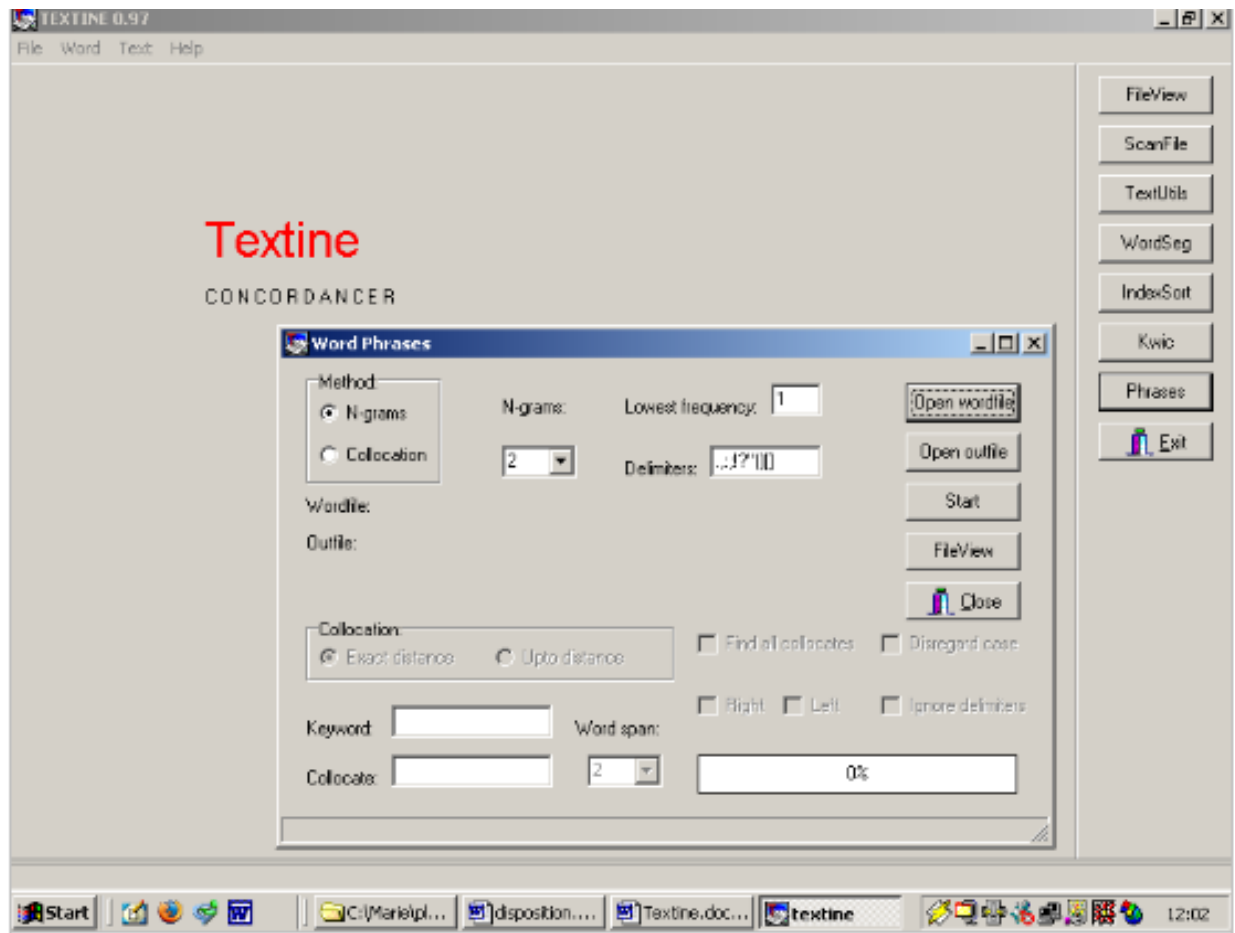
TSSA är ett litet program som är lätt att installera och köra. TSSA startas från TSSA.bat och programmet innehåller många matnyttiga funktioner för språkteknologer, lärare och studenter och andra språkintresserade. Inledningsvis var det dock svårt att veta vilken fil som var själva programmet. Det var stundtals svårt att se vilka filer man själv skapat. De skulle kunna märkas upp eller på annat sätt etiketteras så att användaren själv kan ta bort de filer hon eller han inte längre vill ha. Ibland är det svårt att veta vilken filändelse som ska anges på infilerna. TSSA ger ibland filerna lämpliga ändelser, ibland får användaren själv välja om hon eller han vill ha ändelser. Filer kan även sparas utan ändelser och de går därefter att öppna utan problem. Nackdelen är att användaren inte vet vad det är för typ av sortering filen representerar.

Exempelfiler anges alltid då sådana finns i katalogen så att användaren har något att välja bland. Om filer med korrekt ändelse saknas borde en uppmaning om hur en sådan fil skapas dyka upp på skärmen. Ibland är det svårt att veta hur vissa filer ska skapas; manualen (i pappersform) ger inte svar på alla frågor.

4.3.7 Användarstöd

Användarstödet till programmet är i pappersform och inte heltäckande. För att kunna komma igång räcker dock detta häfte, skrivet på engelska, till. Se källförteckningen för mer information om manualen.

4.4 Textine



4.4.1 Bakgrund

Textine skapades år 2001 av Bengt Dahlqvist vid Institutionen för Lingvistik och filologi vid Uppsala universitet och är en betaversion.¹ Programmet är ett konkordansverktyg som kan behandla information på tecken- och radnivå i en korpus. Textine innehåller funktioner för ordindex, konkordanser, N-gram och kollokationer.

4.4.2 Specifik funktionalitet

Textine har en utvecklad kollokationsbehandling där användaren kan precisera och avgränsa sina nyckelord efter tämligen exakta önskemål.

4.4.3 Metoder och gränssnitt

Konkordansverktyget har åtta funktioner i menyn: FileView, ScanFile, TextUtils, Word-Seg, IndexSort, Kwic, Phrases och Exit. Nedan behandlas de sju första och vad de kan göra med en text.

¹Denna version är en så kallad prerelease och i version 0.976.

1) FileView

Med hjälp av denna funktion tar användaren fram den korpus som ska undersökas.

2) ScanFile

När användaren väljer denna funktion får hon eller han fortsätta att välja en fil att utgå ifrån. Trycker användaren sedan på Char count visas alla tecken som finns i filen med frekvens och ANSI-kod. Det går också att sortera listan efter frekvens. Klickar användaren därefter på Line stats kommer information om raderna i filen fram; vilken som är kortast, längst, totalt antal och antal rader som är kortare eller längre än ett visst antal, av användaren, givna tecken.

I ScanFile finns också en Word length och när användaren klickar på den vill programmet ha en .ina-fil vilken först måste skapas i IndexSort (se nedan).

3) TextUtils

Under denna flik återfinns en samling av funktioner som användaren kan utnyttja för att påverka textsträngarna i sin textfil. Här kan användaren ändra, ta bort, kopiera och lägga till strängar i sin fil. Detta är användbart då textfilen är skriven i en annan kod. Användaren behöver då bara fylla i vad som ska tas bort eller förändras och den nya filen med text får ändelsen .out. Under samma flik finns en funktion kallad More utils och där kan två filer öppnas och användas på tre olika sätt. Merge slår ihop filerna från de två olika input-filerna. Common listar rader som finns i den ena, den andra eller i båda input-filerna. Sentence segment delar upp texterna i rader och det är dessutom möjligt att få versaler utbyta mot gemener.

4) WordSegment

I denna avdelning behövs en textfil och användaren får välja vilka tecken som är ordavgränsande samt vilken typ av ordsegmentering som ska utföras. De filer som genereras är: .wds (för ordindex), .wdk (för kwic-listor), .wdn (för n-gram och meningar) och .wdc för kollokationer. Det går bra att ta med stoppord. Dessa kan plockas ifrån en lista eller så skriver användaren själv in dem i därför avsedd ruta. Textine talar efter segmenteringen om hur många ord som kunde återfinnas i filen.

5) IndexSort

Inputfil här är en textfil, det vill säga korpusen. Det går att välja på fyra olika sorteringsförfaranden (resultatfilernas ändelser inom parentes): att sortera efter ordens första (.ina) eller sista (.fin) bokstav, efter frekvens (.fre) eller efter första förekomsten (.occ). Under respektive förfarande finns fler finesser. Väljer användaren att sortera efter sista bokstaven går det att få ordlistan högercentrerad samt med frekvens. När valet har fallit på ett frekvensförfarande med angivande av procentuell förekomst blir resultatet en fil med fyra kolumner. Den första kolumnen ger ordets frekvens, den andra ger ordets procentuella förekomst, den tredje ger den kumulativa procent-siffran och den fjärde kolumnen är ordet ifråga. Om användaren arbetar med första förekomsten går det att välja att få listan alfabetiskt sorterad. Denna lista ger även information om var i korpusen ordet kan återfinnas.

6) Kwic

Kwic, det vill säga keyword in context, handlar om att få fram ett speciellt ord och hur det samförekommer tillsammans med andra ord. I Textine väljer användaren först hur många ord som ska ingå i konkordansen runt nyckelordet (en lista med givna antal

ges - ju högre siffra, desto större kontext). Konkordanserna sorteras efter förekomst, pre- eller postkontext. Det finns en ruta där det går att klicka i om körningen ska bortse från ord som innehåller andra tecken än bokstäver. Inputfilen är som alltid en korpus, men en .wdk-fil behövs också och den skapas i WordSegment (se ovan). När dessa filer är valda kan programmet köras och resultatfilen är en .wdk-fil. Den består av information om frekvens av en viss konkordans och alla nyckelord står i alfabetisk ordning (dessutom numrerade) med samtliga förekomster av nyckelordet. Utfilen med ändelsen .kwc går att namnge själv. Gör användaren flera körningar skrivs nämligen den gamla filen över. Kwic-sorteringen erbjuder också möjligheten att sortera konkordanserna alfabetiskt efter kontexten före eller efter nyckelordet.

7) Phrases

Under denna flik finns två metoder att tillgå, N-gram och kollokationer, där N-gram är den metod som behöver minst information från användaren. En .wdn-fil är input för denna körning och den skapas i WordSegment (se ovan) och utfilen får ändelsen .ngr. Utfilen består av valt antal ord i n-grammet samt deras frekvens och den är alfabetiskt sorterad.

För kollokationer behövs mer information för en lyckad körning (och dessutom en infil med ändelse .wdc som skapas i WordSegment). Användaren ska ange om kollokationerna ska ha en exakt distans eller hur stort omfång det ska vara. Dessutom är det möjligt att välja om samtliga kollokationer ska tas fram och om de i så fall ska återfinnas till höger eller vänster samt hur många ord som ska ingå i kollokationen. Textine innehåller även möjligheten att körningen ska bortse från om ordet är skrivet med gemener eller versaler. Utfilen innehåller information om på vilken rad de upphittade kollokationerna återfinns.

4.4.4 Prestanda och begränsningar

Textine kräver en dator med Windows (minst Windows 95) installerat. Programmet upptar knappt 800 kB men kräver ledigt utrymme på hårddisken med minst 2 MB. Programmet fungerar med det latinska alfabetet, inklusive å, ä och ö och har dessutom en räknare som beräknar tidsåtgång för en körning. Möjliga infiler till detta program är textfiler. Utfilerna skapas automatiskt av Textine och speglar vad för typ av innehåll filen har, om det exempelvis är en konkordans eller en alfabetiskt sorterad lista.

4.4.5 Test på större texter

Textine tog en halvtimme att ordindexera UNT-92. Den räknade ut att det fanns 6.583.540 ordformer och 313.529 unika ordtyper. Förhållandet (ratio) däremellan redovisades till 20.99. Ovix (ett statistiskt mått på ordvariation) anges till 88.58, vilket är en stor variation. När Textine skulle skapa bigram från textfilen gick det på 1 timme, 13 minuter och 34 sekunder. Filen får då en storlek på knappt 59 MB. Delar av resultatet kan ses i Bilaga C.

4.4.6 Resultat och omdömen

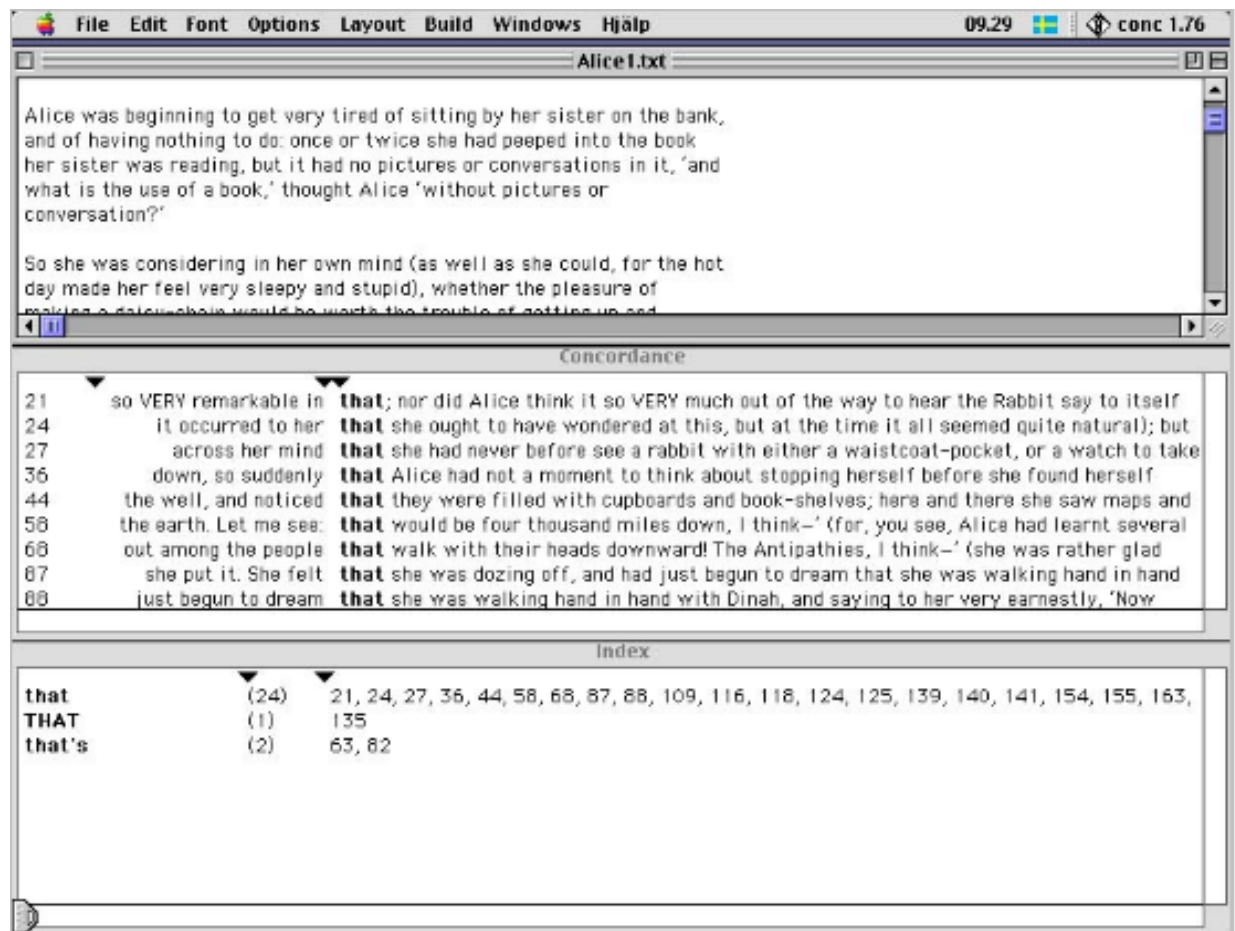
Textine är ett omfattande program med stora möjligheter till olika typer av språkgranskning. Programmet har en räknare som beräknar tidsåtgång för en körning. Ibland verkar den dock inte fungera för räknaren står på 00:00:00 efter misslyckade körningar (vilka rimligen också borde ha tagit tid att utföra).

När man gör körningar på kollokationer så kommer filändelsen .col inte upp automatiskt som ändelserna annars gör i programmet. De måste skrivas in separat, annars går det inte att se filen via FileView. Dock går den att öppna i Wordpad utan problem, trots att filextension saknas.

4.4.7 Användarstöd

Användarstödet till Textine saknas i denna version.

4.5 Conc för Mac



4.5.1 Bakgrund

Detta program är skapat på SIL (Summer Institute of Linguistics) av John Thomson. Denna versions nummer är 1.76 och utkom 1993. Conc har en kortfattad nätmanual som senast uppdaterades 1997. Conc är en RAM-baserad² programvara anpassad för MacIntosh-datorer och kan skapa konkordanser och ordlistor.

4.5.2 Specifik funktionalitet

Conc innehåller några kortare korpusar som användaren kan prova direkt. Vid körning kortar programmet själv av långa rader och anpassar texten till originalstorleken på det öppnade fönstret. Dessutom är det möjligt att med hjälp av reguljära uttryck göra sökningar i korpusen.

²RAM betyder Random Access Memory och Conc använder detta minne för sina körningar. För tidiga versioner av MacIntosh operativsystem kan detta bli en begränsning då körningarna använder mycket minne.

4.5.3 Metoder och gränssnitt

Conc skapar konkordanser och ordlistor. När användaren har skapat dessa kan de ses tillsammans med ursprungskorpusen då användaren klickar på Tile. När ett nyc-
kelord markeras ses sammanhanget i korpusen och samtidigt visas ordet i ordlistan. Under Options är det möjligt att göra inställningar för det man söker, exempelvis ordavgränsare. I Options går det också att välja att inkludera eller exkludera valfria ordformer.

Då samtliga fönster är öppna kan användaren söka i korpusen via tangentbordet. Om ett "A" trycks in kommer det första A:et att markeras på skärmen. Då användaren i snabb följd skriver in exempelvis "Ali" kommer den första strängen med nämnda tre bokstäver att visas.

I statistiken i Build kan man få fram information om sin korpus, exempelvis hur många ord som finns i filen och hur många av dessa som förekommer endast en gång.

4.5.4 Prestanda och begränsningar

Conc för MacIntosh kan bara ha ett dokument öppet i taget. Allt arbete som utförs i programmet ligger i minnet. När användaren har tagit fram det hon eller han är ute efter kan arbetet sparas alternativt exporteras (i File-menyn). Inputfiler är enbart textfiler, men körningarna kan sparas som MS DOS, HTML, RTF, Unicode, Word eller en textfil. Enligt upphovsmannen kan det fortfarande finnas buggar i Conc, men om en användare får problem är det troligast att datorn har begränsat med minne. Det går inte att redigera filer i konkordansfönstret, så om användaren vill gå vidare får hon eller han klippa ut texten och klistra in det i ett annat ordbehandlingsfönster. Däremot går det att skapa en konkordans och utifrån den gå vidare och skapa fler konkordanser. Conc innehåller ursprungligen inte å, ä eller ö, men under Sorting Parameters finns ett fönster där det går att lägga till nämnda bokstäver. Dessutom kan användaren kan byta typsnitt och storlek om så önskas.

4.5.5 Test på större texter

Den MacIntosh-dator som användes vid testet kunde inte läsa av UNT-92 från CD-skivan, varför denna punkt får lov att utgå.

4.5.6 Resultat och omdömen

Conc är ett ganska enkelt program med basala funktioner inom konkordans- och ordlistearbeten. Då en mindre korpus används levereras resultaten snabbt, räknare saknas dock. Till det positiva med detta program hör möjligheten att kunna söka med hjälp av reguljära uttryck. Det kan vara mycket användbart i olika språkteknologiska sammanhang. Användarskaran kan dock tänkas vara begränsad då MacIntosh-systemen inte är så utbredda.

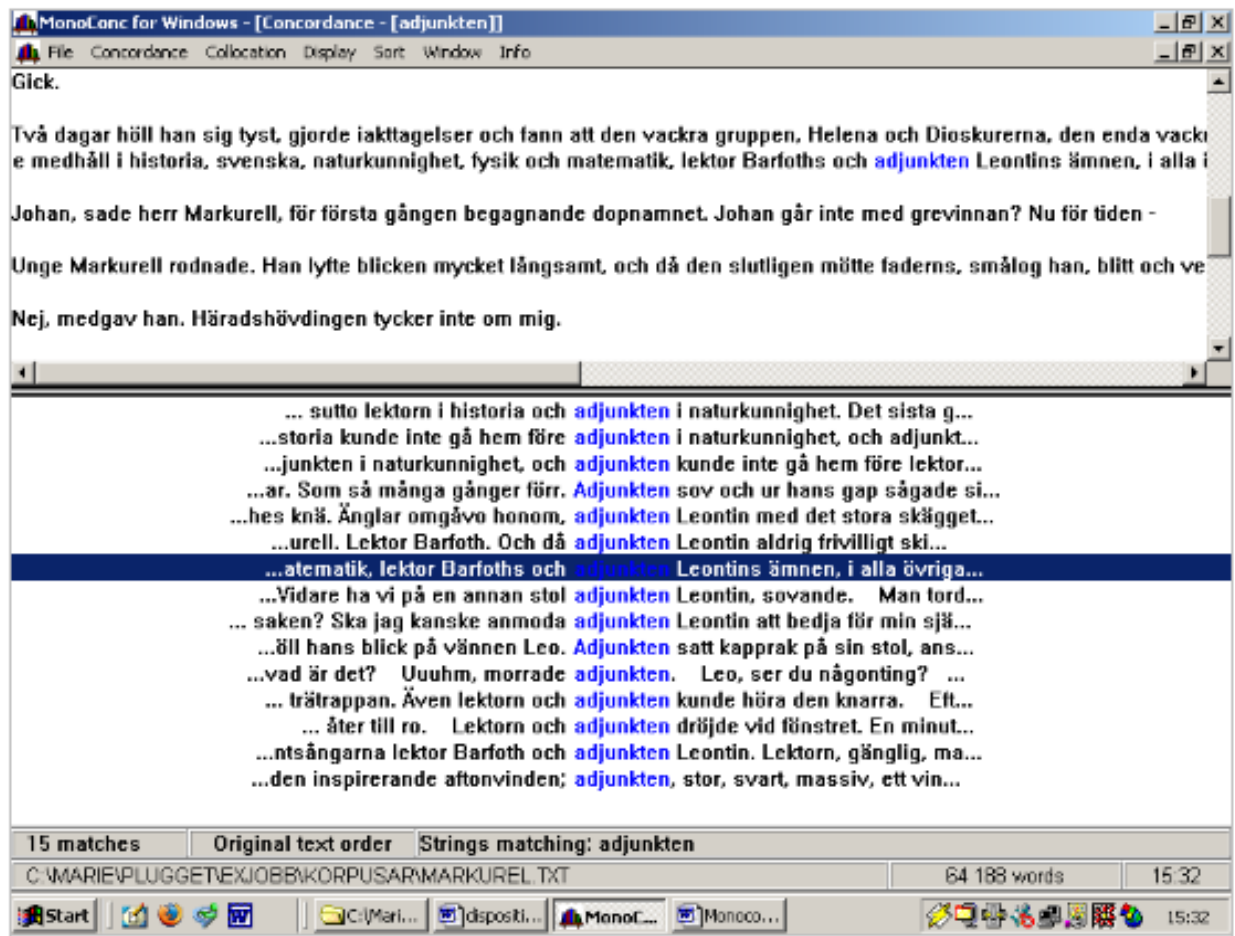
4.5.7 Användarstöd

Handledningen³ består av tre delar: en nybörjarnivå, en mellannivå och en avancerad nivå. Samtliga tre delar är lättfattliga och det är inga problem att förstå dem. Nackde-

³En så kallad Tutorial finns att hämta på <http://www.sil.org/computing/conc/tutorial.html>

len är att det inte går att söka efter ett speciellt problem; handledningen måste läsas från början till slut. I den slutliga delen behandlas reguljära uttryck och är det så att användaren inte stött på detta förut kan kunskaper behöva inhämtas i detta område innan den sista delen läses.

4.6 Monoconc



4.6.1 Bakgrund

Den version av Monoconc som har undersökts är version 1.1, skapad 1996 av Michael Barlow (och programmerad i Borland Delphi av Steve Neumann). Monoconc föreligger i fler versioner efter denna, bland annat Monoconc 2.0 och Monoconc Pro 2.2, båda med utökade finesser jämfört med den version som här testats.⁴

4.6.2 Specifik funktionalitet

I Monoconc kan samtliga konkordanser läsas och jämföras simultant med kontexten för konkordansen, tack vare ett tudelat fönster. I programmet finns även möjligheten att ändra typsnitt och textfärger.

4.6.3 Metoder och gränssnitt

Monoconc körs i ett fönster med en menyrad med ett antal olika val för användaren: Corpus Text, Concordance och Collocation. Under Corpus Text är det möjligt

⁴källa: <http://www.camsoftpartners.co.uk/monoconc.htm>

att byta typsnitt och om filen ifråga endast består av en lång rad ord (som försvinner ut till höger i fönstret) kan användaren klicka på Word Wrap så återställs texten. Under Concordance kan användaren med hjälp av Search söka efter ett mönster med reguljära uttryck.⁵ När programmet har gjort en sökning delas korpusfönstret i två delar. Den övre delen visar kontexten för det sökta uttrycket, det nedre fönstret visar resultatet av sökningen. Genom att klicka på ett av uttrycken i det nedre fönstret visas kontexten i det övre. När Monoconc gjort en sökning utvidgas valmöjligheterna i Search-menyn till att omfatta bland annat tilläggsökning (Search append), möjlighet att spara en fil eller stänga ner den. Redan tidigare fanns görligheten för användaren att modifiera sin sökning, exempelvis genom att bestämma sökparametrar såsom maximalt antal träffar, bortse från gemener/versaler, val av ordavgränsare med mera. När en konkordanslista tagits fram kan användaren gå in i menyn Sort och där sortera efter fem olika alternativ: sortering efter första eller andra ordet till vänster eller höger samt efter sökordet. Det går också att ta tillbaka ordföljden som den ser ut i originaltexten. Vill användaren av någon anledning radera rader går det att göra i menyn Display (alternativt Ctrl-D) efter att raden markerats. I Display finns också möjligheten att ändra typsnitt och sökordstextfärg (default är blå färg).

Under Collocation finns frekvensdata för korpusen. Det är möjligt att få frekvensdata för kollokationer eller korpus sorterad enligt frekvens eller alfabetiskt. För att få en lista över kollokationer måste sökning på ett ord i korpusen ha skett först. Båda sorteringarna ger upphov till en lista med tre kolumner. I den första återfinns antal förekomster, i den andra den procentuella andelen av hela texten och i den tredje finns ordet. Användaren kan dessutom själv modifiera sin sökning. Monoconc kan ställas in efter lägsta frekvens av ett valt ord, maximalt antal rader, om körningen ska ta med alla ord eller bara innehållsord. En stoppordlista kan också skapas genom att för hand skriva in de ord som önskas.

Under Sort finns de alternativ som även kan användas vid kollokationssortering. Se förfarandet under Concordance.

4.6.4 Prestanda och begränsningar

Monoconc är ett litet program som upptar 406 kB och det körs i Windows. Programmet saknar en räknare för hur lång tid en körning tar. Det latinska alfabetet, inklusive å, ä och ö går att använda. Monoconc efterfrågar enbart textfiler som möjliga infiler. Utifilerna sparas inte automatiskt. Användaren får själv gå in och spara (valet finns i fönstermenyerna) och samtliga filer som skapas blir textfiler.

4.6.5 Test på större texter

Programmet laddar in UNT-92 på drygt sex minuter och visar därefter att det finns 6.600.194 ordformer i filen. Då jag söker på ordet "journalist" får jag 123 träffar och det tar knappt två minuter för denna sökning.

4.6.6 Resultat och omdömen

Textmassan som visas i det övre fönstret vid konkordanssökning (klicka på vald konkordans så visas textmassan) går ej att förändra längdmässigt. Det antal tecken som

⁵De reguljära uttryck som finns att laborera med här är: %, *, ? och @.

visas före och efter nyckelordet är standard. Om användaren inte innan gjort en Word Wrap kan det vara svårt att hitta det sökta nyckelordet i kontexten.

Det är viktigt att komma ihåg att fortsatta konkordanssökningar ger fler svar i den undre rutan, som hela tiden byggs på. Körningarna löper på ända tills man stänger fönstret (under Corpus Text) och gör en ny körning alternativt raderar fönstrets innehåll, rad för rad, vilket kan vara omständligt om träfflistan är lång. Längst ner i fönstret står vilken typ av sökning som gjorts och hur många träffar som åstadkomits. Det är bra då användare kanske inte alltid minns sitt senaste val.

Programmet är känsligt för gemener eller versaler. Detta är till programmets nackdel då användaren till exempel vill söka på uttrycket ”för så vitt”. Ibland inleder frasen en mening vilket en konkordanssökning kan hitta. Vill användaren sedan gå vidare och undersöka kollokationer i konkordanserna tas ej de fraser som innehåller inledande versal med. Är man inte observant på detta kan resultatet bli missvisande.

Monoconc kan förutom det rent språkliga även förändras estetiskt. Olika typsnittsalternativ och teckenfärg finns att välja på.

4.6.7 Användarstöd

Användarstödet till Monoconc är en hjälpfil som kan nås i fönstermenyn. Hjälpen till användaren är enkel och informationsrik.

4.7 IMS Corpus Workbench

4.7.1 Bakgrund

IMS Corpus Workbench är ett tyskt program, skapat vid universitetet i Stuttgart av Oliver Christ, Bruno M. Schultze, Anja Hofmann och Esther König. Programmet Corpus Query Processor (CQP) som ingår i IMS Corpus Workbench är avsett att användas för olika typer av konkordanssökningar och ordindexskapande. Den version som har undersökts är version 2.2 från 1999, (Christ, 1999), men det finns nyare versioner med fler finesser. CQP skapar konkordanser och klarar av sökningar med reguljära uttryck.

4.7.2 Specifik funktionalitet

IMS Corpus Workbench kan hantera taggade korpusar och är avsett för mycket stora korpusar.

4.7.3 Metoder och gränssnitt

För att starta CQP behöver användaren ett terminalfönster och kunskaper om hur frågor, sk queries, till programmet görs. För att starta programmet skriver man in "cq" och för att avsluta det "exit;". Alla kommandon till CQP ska avslutas med ett semikolon. För att ta reda på vilka korpusar som finns i systemet kan användaren skriva "show;". Därefter väljs önskad korpus och sökningar kan göras. Om användaren exempelvis vill se hur många träffar som genereras på ordet "springa" så skriver hon eller han in följande:

```
SUC> ''springa'';
```

samt klickar på Enter. Ur SUC-korpusen genereras då samtliga förekomster av ordformen "springa". Om träffarna blev alltför många kan användaren göra om sökningen och lägga till en "cut" med önskat antal träffar, exempelvis:

```
SUC> ''springa'' cut 10;
```

Här visas då de första tio träffarna som programmet genererar.

När dessa sökningar görs visas radnummer och nyckelordet i sin kontext. Nyckelorden är centrerade och inom hakparenteser.

Det är möjligt att via kommandorader bestämma hur många tecken som ska finnas till höger eller vänster om nyckelordet. Vill användaren begränsa kontexten till antal ord ska detta specifikt anges. Nedan visas exempel på begränsningar i tecken- respektive ordkontext.

```
SUC> set LeftContext 10;  
SUC> set RightContext 10;  
SUC> set Context 10 word;
```

Det går att skicka konkordansresultaten till valda filer genom att använda så kallade pipes. På samma sätt är det möjligt att skicka sina resultat till en skrivare. Från version 2.3 är det möjligt för CQP att producera resultat direkt i LaTeX-kod.

Vill användaren se beräknad tidsåtgång för en sökning måste kommandot för detta skrivas in:

```
SUC> set Timing yes;
```

Reguljära uttryck är mycket användbara i konkordanssammanhang. Användaren som vill söka efter ett specifikt ord, där ordet ifråga kan förekomma med både gemen och versal inledningsvis, kan använda följande kommando:

```
SUC> '(t|T)idningen';
```

För mer information om reguljära uttryck och deras användning hänvisas till susning.nu:s hemsida⁶.

4.7.4 Prestanda och begränsningar

CQP körs på en Linux-dator och till sin hjälp bör användaren då ha en editor (exempelvis emacs) och ett terminalfönster.

När man gör en enkel sökning visas inte antalet träffar. Radnummer visas och nyckelordet är centrerat. Det är möjligt att använda samtliga alfabeten. Om användaren har tillgång till en ordklasstaggad korpus finns mängder av finesser att tillgå. Se vidare i manualen till IMS Corpus Workbench (Christ, 1999).

4.7.5 Test på större texter

UNT-92 kunde inte läggas in i det system som behövdes för att IMS Corpus Workbench skulle kunna läsa filen. Enligt upphovsmännen är den största möjliga korpus som hittills körts i programmet en tysk tidningskorpus på cirka 200 miljoner ordförekomster annoterade med lemman, två olika pos-tagset och meningsavgränsare.

4.7.6 Resultat och omdömen

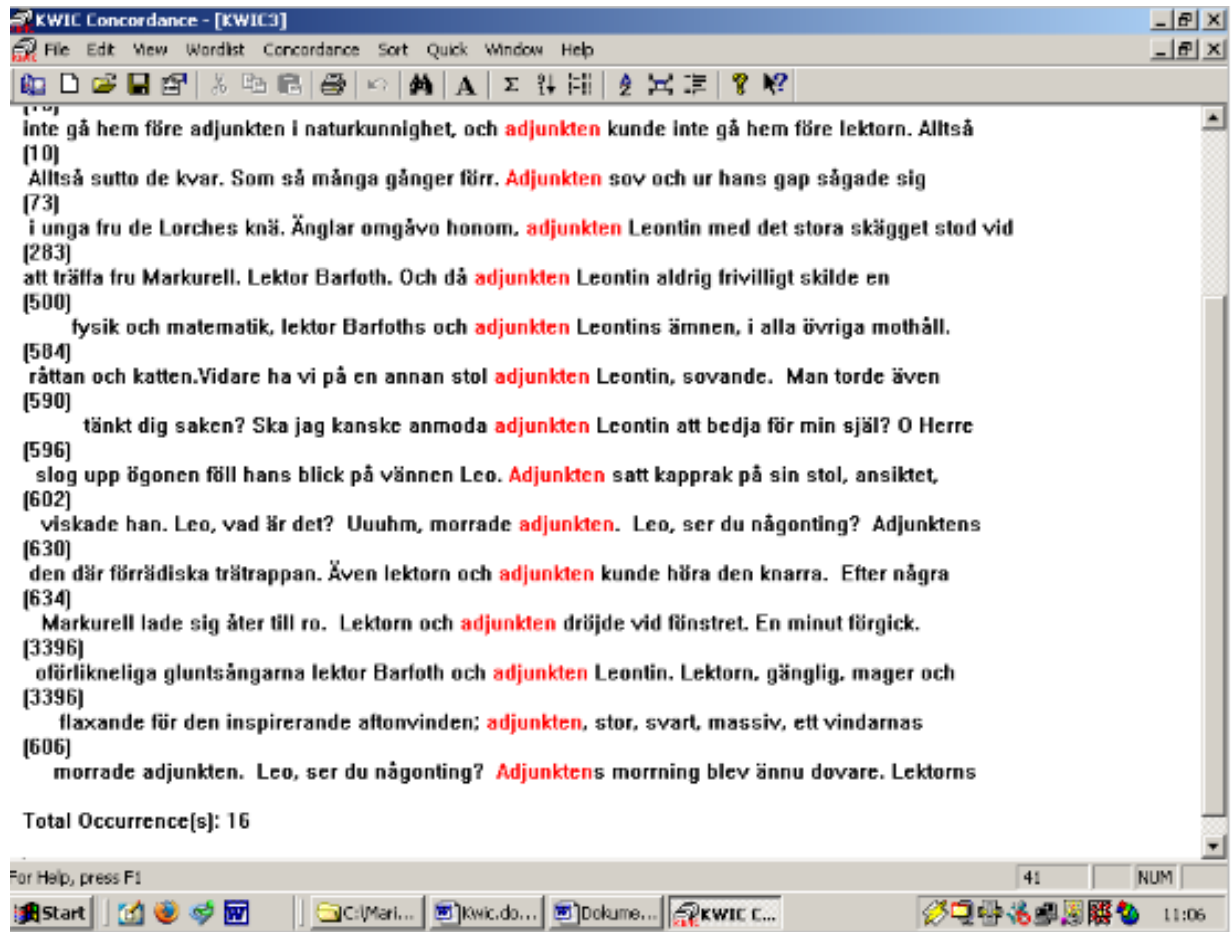
CQP är ett omfattande verktyg, redan i den version som har provats. Programmet började utvecklas 1993 och har erhållit fler finesser under tidens gång. Det som gör CQP aningens svårarbetat är arbetet som måste ske initialt - att lägga upp en korpus. I manualen till CQP anges att systemadministratören är den person som är bäst lämpad att utföra detta arbete. När programmet väl är installerat är det enkelt och lättarbetat.

4.7.7 Användarstöd

Manualen till IMS Corpus Workbench (Christ, 1999) är skriven av några av upphovsmännen till programmet och är mycket omfattande, men inte svårhanterlig. Vissa punkter på dagordningen hänvisar till en nyare version av programmet, men det får man bortse ifrån.

⁶http://susning.nu/Regulj%E4rt_uttryck

4.8 Kwic



4.8.1 Bakgrund

Kwic-versionen som har körts är 4.7 och är skapad av Sakoru Tsukamoto 2004. Programmet kan skapa ordlistor, konkordanser och kollokationer baserade på korpusar som användaren väljer.

4.8.2 Specifik funktionalitet

En av de funktioner som är unik med Kwic är att det är möjligt att ta fram ett index för varje ordtyp. Programmet listar ordet eller siffran samt alla de rader som ordtypen förekommer på. Kwic har dessutom ett rimindex där alla ord som står i slutet av konkordansraderna är medtagna. Denna möjlighet finns under WordList.

4.8.3 Metoder och gränssnitt

Då verktyget startas öppnas ett fönster där användaren först måste välja en korpus och därefter justera inställningarna i densamma (under Corpus Options). Kwic har ett menysystem där användaren kan gå in och välja att arbeta med ordindex eller konkordanser.

Under Wordlist finns följande alternativ att tillgå:

- Wordlist - en ordlista inklusive frekvens
- Backward List - sorterad efter sista bokstaven, inklusive frekvens
- Descending Wordlist - frekvenslista, alfabetiskt sorterad
- Ascending Wordlist - frekvenslista, alfabetiskt sorterad
- Index - med radnummer för varje ordförekomst
- Rhyme Index - ett rimindex

Varje körning avslutas med att programmet uppger hur många ord och ordtyper som ingår i filen. Under Concordance finns följande alternativ att tillgå:

- Kwic Concordance - listar nyckelord efter förekomster, inklusive radnummer
- Kwic Concordance sort by Left - listar nyckelordet efter ordet till vänster
- Kwic Concordance sort by Right - listar nyckelordet efter ordet till höger
- Collocation - listar alfabetiskt ord som förekommer med nyckelordet
- Concordance All - listar alla konkordanser alfabetiskt
- Kwic format - användaren väljer hur utfilen ska se ut (om orden ska vara separerade av tabbar, komma eller om orden ska vara taggade, exempelvis i xml)

Nyckelordet rödmarkeras alltid i resultatfilen. Varje körning avslutas med att programmet uppger hur många förekomster av nyckelordet som återfinns i filen. Under Sort kan användaren sortera sin senast körda ordliste- eller konkordansfil (från korpussen). De konkordanser som tidigare varit rödmarkerade försvinner. Har användaren skrivit in ett annat nyckelord rödfärgas det inte heller. En alfabetisk ordlista som tidigare körts kan ändras till en frekvensordlista och tvärtom.

Quick fungerar på ett liknande sätt. De nya filerna baserar sig inte på korpussen utan det tidigare Kwic-fönstrets innehåll (inklusive filnamnet och annan information som finns i fönstret). Det går att manipulera orden i fönstren. Det är även möjligt att göra ordlistor av konkordansfilen. Ur konkordansfilen går det att ta fram nya konkordanser (som även de är rödmarkerade).

4.8.4 Prestanda och begränsningar

Kwic körs i Windows och tar upp 728 kB på hårddisken. Programmet klarar det latinska alfabetet samt reguljära uttryck Genom att söka på strängen "are" i en konkordans kommer samtliga ord i korpussen innehållande dessa tre bokstäver i nämnd ordning att tas med i utfilen.

Möjliga infiler är textfiler, olika korpustaggade filer (BNC, COCOA, LOB, Brown med flera). Däremot sparas alla filer i ett .kwic-format.

4.8.5 Test på större texter

Kwic klarar att ladda in UNT-92-korpusen på ungefär sju minuter. Sedan tar det ungefär ytterligare 85 minuter för programmet att sammanställa en alfabetiskt ordnad frekvensordlista. Programmet redovisar att det finns 6.632.428 ordformer i korpusen och att det är 290.064 olika ordtyper. Då jag gjorde en konkordanssökning på ordet "Uppsala" visades 18.982 förekomster och denna sökning tog 16 minuter.

4.8.6 Resultat och omdömen

Kwic är lättöverskådligt tack vare sitt menysystem. Programmet upplevs dock långsamt, även för mindre korpusar. Tidsangivelser saknas för körningar. En engelsk korpus går utmärkt att köra, men en svensk kan göra att det blir problem i och med våra avslutande bokstäver i alfabetet. Problemet uppstår då Windows 2000 i engelsk version används. För att lösa detta kan användaren i Kwic gå in under File och Corpus Options. Där finns möjligheten att välja vad infilen ska vara för typ (olika taggssystem för korpusar går bra, till exempel) och vilket språk infilen avser. När programmet skapar ordlistor i alfabetisk ordning sorteras siffror före själva alfabetet.

När konkordanser körs blir nyckelorden rödmarkerade vilket gör filen lättläst. Tyvärr så listas inte nyckelorden exakt under varandra. Det ger ett något rörigt intryck.

Under "Concordance" finns en undermeny där det står "Sort by left/right". När användaren sorterar enligt vänster står orden till vänster om nyckelordet i alfabetisk ordning (undantag om en genitivpostrof återfinns framför ordet - då räknas enbart "s" som ett ord). När en körning som sorterar enligt höger görs ser det ut på följande sätt då "Hamlet" skrivs in som nyckelord: (klippt ur konkordanslistan)

```
(527)
father bears his
(894)
father comes.
(478)
father in the
(505)
father lost
(506)
father lost,
(268)
father lost:
(578)
father Than I
(662)
father! Hor.
(574)
father's body
(578)
father's brother;
(623)
father's funeral.
(452)
father's leave?
```

(794)
father's person,
(813)
father's spirit
(677)
father, Armed
(407)
father, with all
(637)
father,--methinks
(31)
Father. Gertrude,
(659)
father. Ham.
(637)
father. Hor.
(439)
father. What
(524)
father: for let
(504)
father; But, you
(505)
father; That
(689)
father; These
(520)
fathers, and who

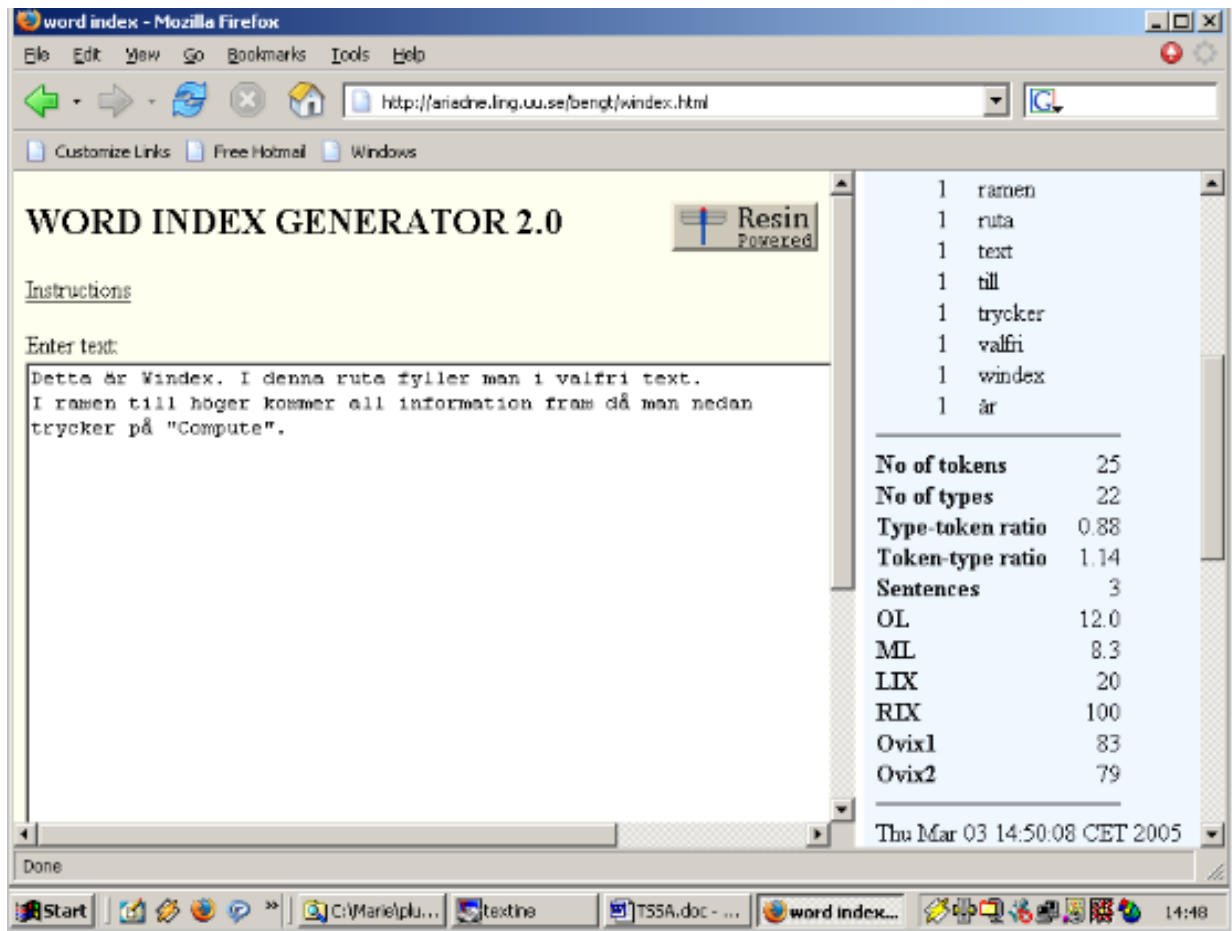
Här är det vid en snabbgranskning inte så enkelt att se hur programmet har sorterat. Det är tydligt att en åtskillnad görs på olika skiljetecken, gemener och versaler förutom det rent alfabetiska. Frågan är om användaren vill ha sorteringen på detta sätt. Det finns inte någon möjlighet för användaren att gå in och styra sorteringen.

Ett rimlexikon ska kunna skapas ur korpusfilen. Om användaren väljer detta kommando körs programmet och ord som står i slutet på alla rader listas. Frekvens- och radnummer följer på ordet ifråga. Det går inte att söka på rader i Kwic, så denna möjlighet är inte lätt för användaren.

4.8.7 Användarstöd

Under Help Topics finns inte mycket hjälp gällande rimlexikonskapande att få. Användarstödet ser vid en första anblick gediget ut, men då hjälp till ett slumpartat problem söks är beskrivningen relativt kortfattad. För övrigt är det en del stavfel och andra småfel i Help Topics; programmet känns inte riktigt färdigarbetat med avseende på denna del.

4.9 Windex



4.9.1 Bakgrund

Detta ordindexverktyg kommer från Bengt Dahlqvist vid Institutionen för Lingvistik och filologi vid Uppsala universitet. Programmet skapades 2001 och går att nå på webben. Word Index Generator (Windex) bygger på samma källkod som TSSA (se sektion 4.4). I ramen till höger framkommer statistik om den inskrivna texten.

4.9.2 Specifik funktionalitet

Detta program har sin tyngdpunkt på textens statistik. Användaren får reda på hur pass läsbar en text är. Texten behandlas bland annat med avseende på följande: uppskattning av ordvariation, procentuell andel "svåra" ord, medellängd per mening och läsbarhetsindex, så kallad LIX.

4.9.3 Metoder och gränssnitt

Windex är ett program som körs direkt på webben. Användaren skriver in text (alternativt utnyttjar funktionen klippa/klistra) i en ruta direkt i webbläsaren. I ramen till höger framkommer statistik om den inskrivna texten då användaren klickar på "Compute". Det finns även den möjligheten att utfallet av en körning kan justeras. Det går

att bestämma vilka skiljetecken som ska användas samt vilket språk programmet ska sortera enligt (engelska, svenska, franska eller tyska). Dessutom kan programmet konvertera alla versaler till gemener samt tillhandahålla en frekvenssortering. I ramen till höger framkommer en alfabetiskt sorterad lista efter en körning. Denna lista inkluderar även ordfrekvenser. Under listan ges statistik kring den inskrivna texten. Här anges totalt antal ord, antal ordtyper, uppskattning av ordvariation, antal meningar, procentuell andel svåra ord (ord längre än sex tecken), medellängd per mening och läsbarhetsindex (LIX). Flera varianter av samma typ av statistik förekommer.

4.9.4 Prestanda och begränsningar

Enligt upphovsmannen tar en körning med en text på runt 100.000 ord fem sekunder att genomföra. Programmet är inget egentligt konkordansprogram utan mer ett verktyg för att skapa ordindex, såsom programnamnet antyder. Det latinska alfabetet är det alfabet som går att använda, inklusive å, ä och ö. De filer som skrivs (eller klistras) in ska vara ren text. Det går inte att söka efter en text via någon browse-funktion. Utfilen, resultatet, finns i en ram till höger. Där återfinns all statistik.

4.9.5 Test på större texter

Windex klarar inte en så stor mängd text som UNT-92 utgör. Både Windows-datorn och Linux-datorn hängde sig vid försöken.

4.9.6 Resultat och omdömen

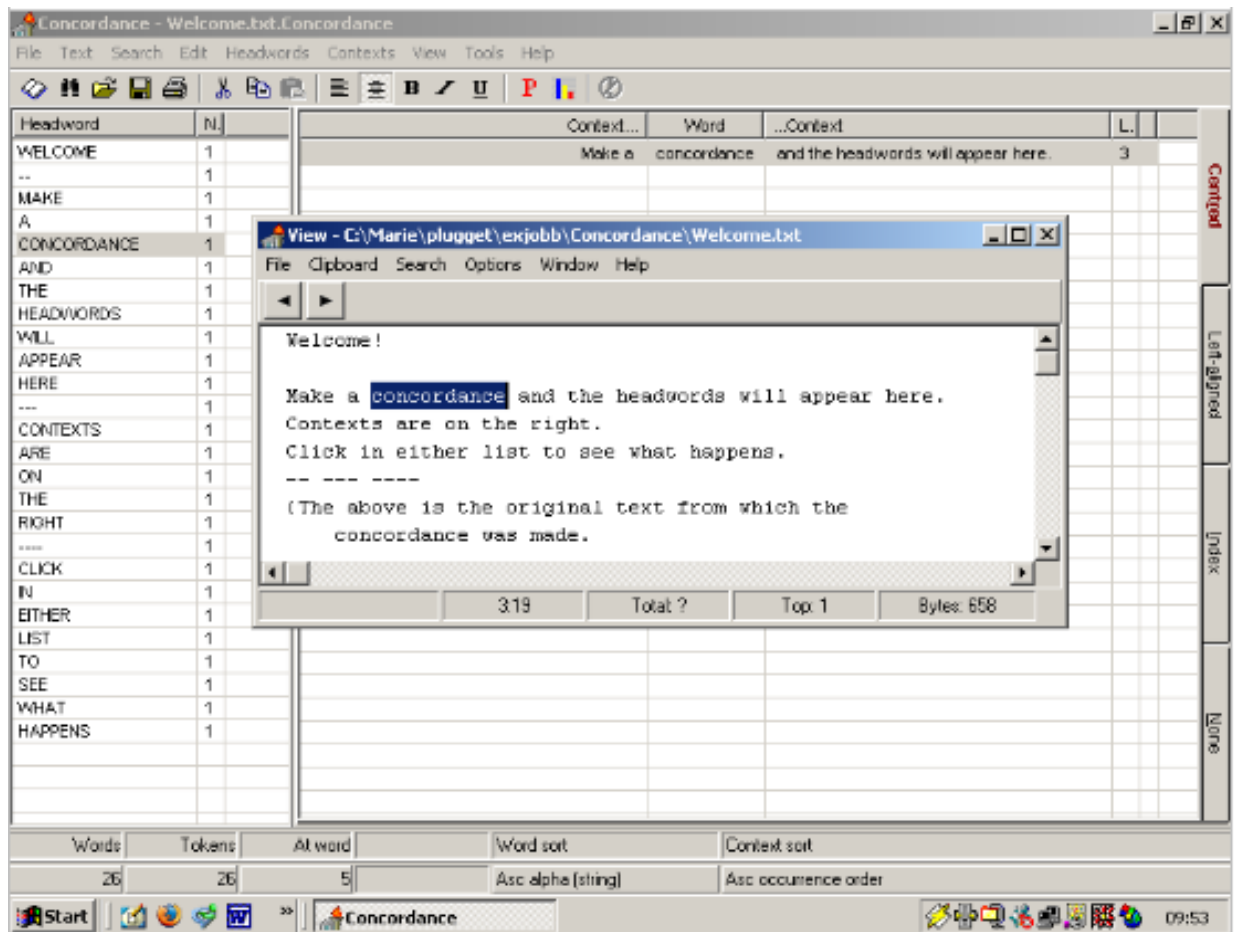
Windex är ett lättillgängligt och lättförståeligt program. Det saknas uppgifter om hur fort en körning går att genomföra, men det är inte av någon större betydelse då användaren sällan skriver in större textmängder. Körningarna går mycket snabbt att utföra. Detta enkla lilla program ger användaren mycket information om den inskrivna texten.

4.9.7 Användarstöd

Användarstödet består av en länk på Windex sida (Instructions). Instruktionerna består av en sida text, vilken är lättförståelig. Dock bör användaren ha lite bakgrundskunskaper om vad till exempel LIX⁷ är samt ha en viss förståelse för statistiska mått.

⁷Mer information om LIX finns att läsa på <http://sv.wikipedia.org/wiki/LIX>

4.10 Concordance



4.10.1 Bakgrund

Concordance tillkom 1999 och upphovsmannen till detta program är R.J.C. Watt. Programmet är ett konkordansverktyg som innehåller funktioner för ordindex, kollokationer och konkordanser. Dessa är möjliga att beskåda simultant.

4.10.2 Specifik funktionalitet

Med Concordance hjälp kan användaren skapa konkordanser och lägga ut dem som fullständiga HTML-filer på nätet. Programmet har också den möjligheten att användaren kan lemmatisera en ordlista genom att gruppera önskade ord.

4.10.3 Metoder och gränssnitt

Programmet går igenom filen när användaren laddat in önskad korpus via menyradens ”Make Fast (eller Full) Concordance”. Om programmet finner raderna för långa skrivs detta ut (”Better re-format the file”) på tillämpliga ställen. En körning avslutas med att ”Progress Dialog”⁸ i Concordance talar om tre saker: hur lång tid det tog

⁸Denna dialogruta kan tas fram under ”View” närhelst användaren behöver information från denna.

att analysera filen samt tidsåtgången för sortering och konkordans. Då körningen är avslutad återfinns en alfabetisk ordlista, inklusive frekvens, i den vänstra ramen. Den ordform användaren vill gå vidare och analysera är klickbar i vänsterramen och konkordansen visas i sin kontext i höger ramen, det stora fönstret. Då användaren klickar på noden, det vill säga nyckelordet, dyker en ruta upp vilken återger hela korpusen och nyckelordet är markerat däri. I den vänstra ramen finns en klickbar del, headword. Efter en första körning ses alla ordformer i alfabetisk ordning. Listan inleds med tecken såsom citationstecken, nummertecken och bindestreck. Därefter följer siffror och sedan kommer själva orden. När användaren klickar på headword ändras listans utseende. Användaren ser längst ner i mitten av fönstret vad det rör sig om för en listtyp. Listorna presenteras i följande ordning⁹:

- asc alpha (string)
- desc alpha (string)
- asc frequency
- desc frequency
- acc length
- desc length
- word endings (word)
- word endings (string)
- occurrence order
- asc alpha (word)
- desc alpha (word)

I det stora fönstret, kontextfönstret, finns några olika möjligheter att undersöka ett konkordansresultat (dessa val är även möjliga att nå från menyraden). Default är att nyckelordet är centrerat, men det är möjligt att få konkordansen vänstercentrerad samt att ta fram information om vilken rad i korpusen nyckelordet återfinns på.

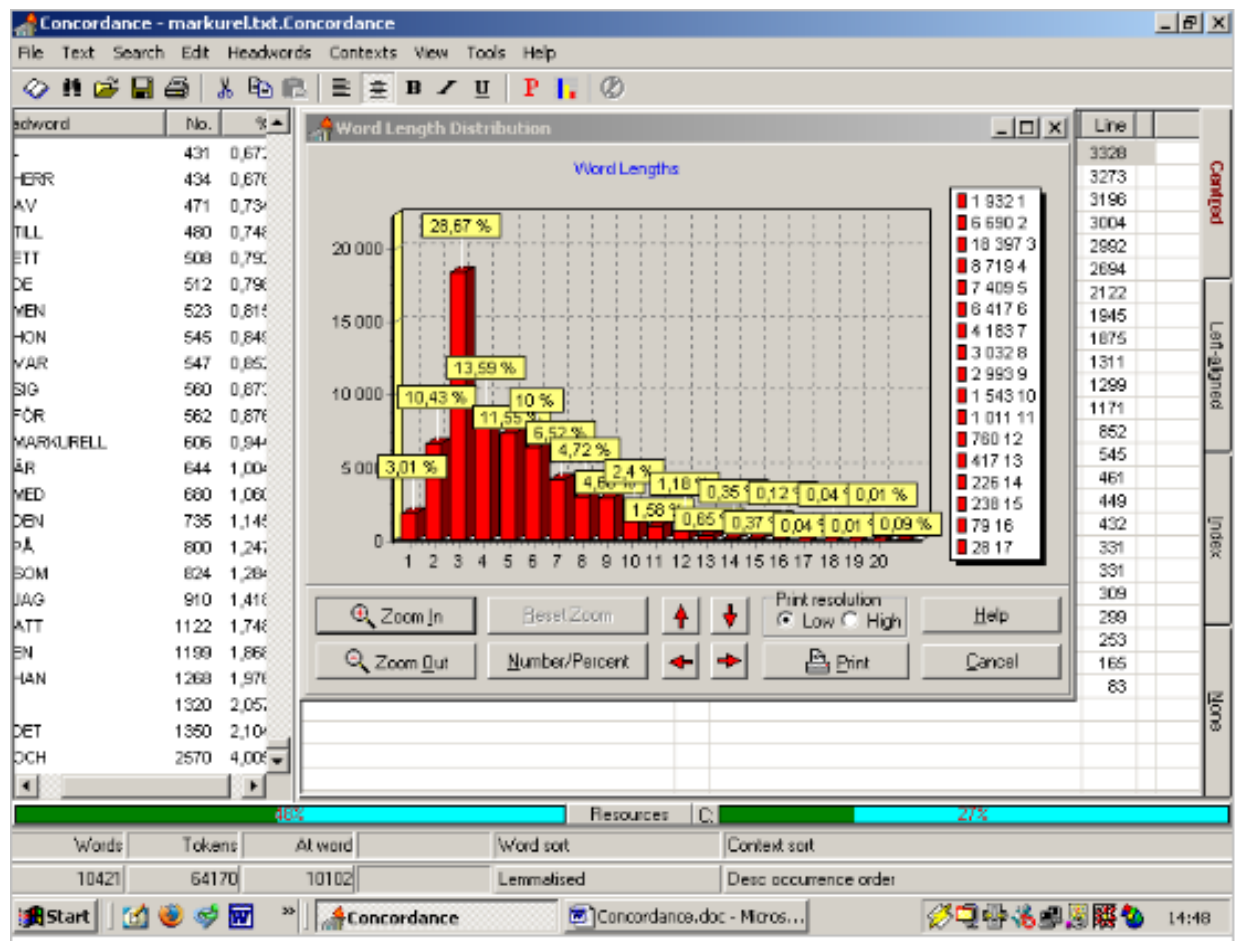
I menyraden finns ett rött P som står för Properties, egenskaper. Då denna bokstav klickas ses en ruta med allmän information om den laddade korpusen samt den senaste sökningen. Den allmänna informationen består av vad källfilen heter, hur många rader den innehåller, antal ord och ordformer i korpusen, ordform/ord-proportion, antal tecken, antal meningar och antal ord per mening i genomsnitt.

I programmet finns en gedigen sökfunktion, under "Search" i menyraden. Här kan användaren söka efter ett huvudord (exakt matchning) eller huvudord som inleds på ett visst sätt. Kontextfönstret är också sökbart.

Under "Headwords" i menyraden finns ytterligare ett antal valmöjligheter att arbeta med, nämligen kollokationer och lemmatisering. Kollokationerna visas i ett separat fönster liksom lemmatiseraren. Den senare innehåller initialt ett antal engelska ord.

⁹asc betyder stigande och desc betyder fallande

I "View" kan användaren exempelvis se en färgglad och informationsrik tabell, vilken kan redigeras efter önskemål, över ordlängderna, se exempel nedan.



4.10.4 Prestanda och begränsningar

Concordance, som körs i Windows, kräver ett utrymme på 3,74 MB på hårddisken samt ett stort minne för att klara av ansenliga korpusar. Enligt Watt begränsas storleken av körbara korpusar enbart av tillgängligt minne på datorn samt hårddiskutrymme. Detta program är avsett för Windows, men även en Mac-användare kan nyttja programmet med hjälp av en pc-simulator. Dock går körningarna mycket långsammare enligt upphovsmannen. Watt exemplifierar med en körning på en 600 MHz Pentium III-dator: Concordance kan då plocka ut 15.000 förekomster av ett ord ur en 1.5 MB stor text på under fyra sekunder.

Flera inputfiler är möjliga att använda simultant. Konkordanser kan skapas från textfiler eller om användaren så vill, från HTML-filer. Det är möjligt att enkelt skapa webkonkordanser för publicering direkt på nätet.

Concordance innehåller två inbyggda program. En File Viewer som kan visa hela korpusen och en fileditor som kan hantera redigering av filer upp till 16 MB. Det finns också verktyg för att konvertera från OEM-¹⁰ till ANSI-tecken och från Unix- till pc-

¹⁰Original Equipment Manufacturer innebär att en datortillverkare köper utrustning från en annan tillverkare och sedan säljer den som en del av ett system som säljs under eget namn.

filer. Programmet klarar av samtliga alfabet och teckenuppsättningar. Användaren kan själv enkelt gå in i en editor och definiera sitt alfabet.

4.10.5 Test på större texter

Under inladdningen av UNT-92-korpusen lägger programmet till alla icke-alfabetiska (@, \$, ê etc) tecken, vilket kan ses i ett fönster som visar inladdningens fortskridande. Concordance analyserade källtexten på en minut och 15 sekunder. Under sorteringsprocessen skapas en mängd temporära filer som fyller upp hela skrivbordsytan, men som sedan automatiskt raderas då programmet sorterat färdigt. Concordance sorterar korpusen enligt fallande alfabetisk ordning på knappt 15 minuter. Korpusen laddas därefter upp på ungefär en halvtimme och antal ord anges vara 6.593.020. En sökning på ordet journalist ger 123 träffar, men söktiden uppges inte.

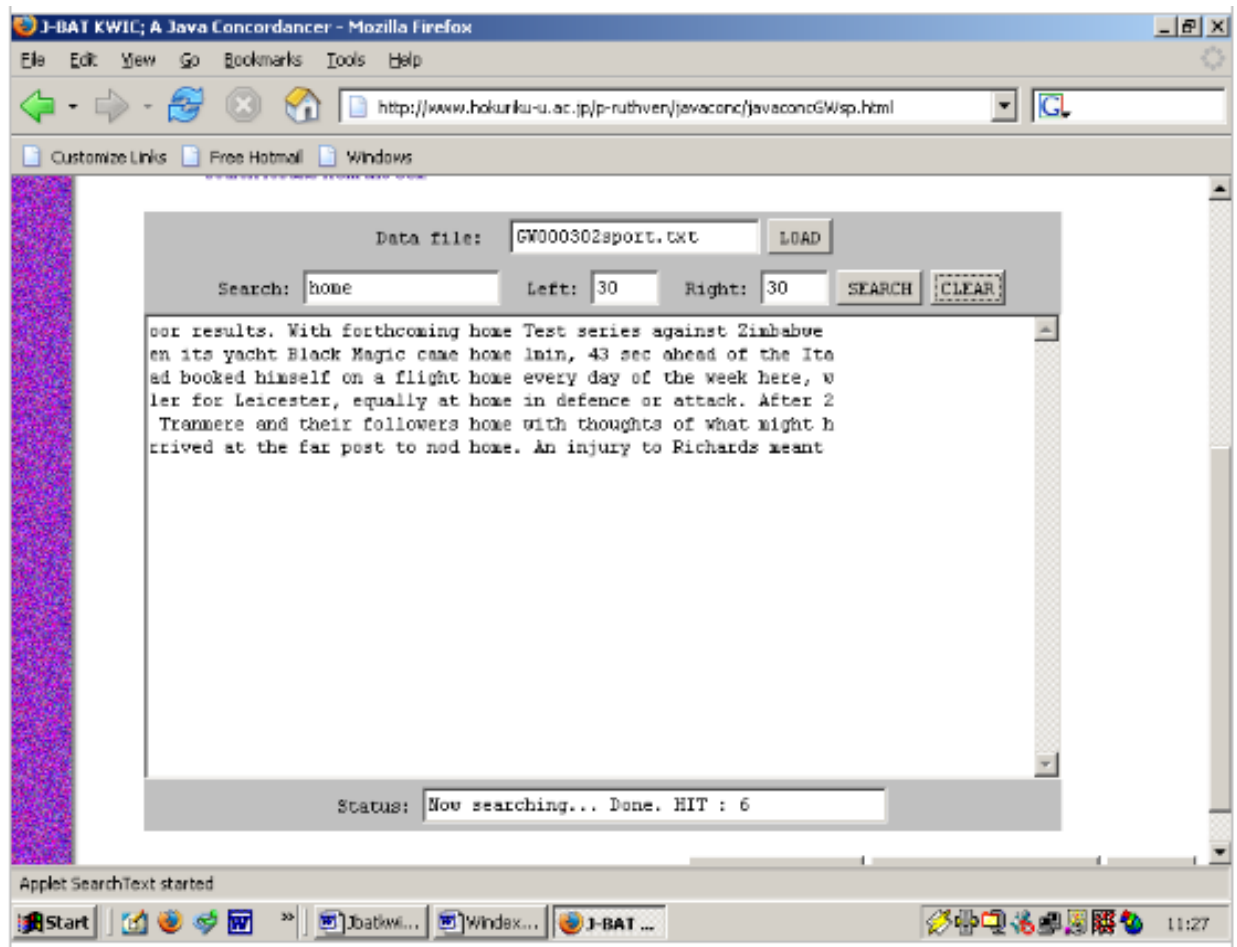
4.10.6 Resultat och omdömen

Concordance är ett omfattande program, användarvänligt och robust. Jag har inte stött på några buggar eller svårhanterliga problem under undersökningens gång. Det är enkelt och överskådligt och lätt att hoppa mellan ordlistan konkordansen och korpusen under arbetets gång. Då användaren önskar söka efter kollokationer är förfarandet inte så enkelt och intuitivt. Ett nytt fönster visas på skärmen med ett antal ord som förekommer till höger och vänster om nyckelordet. Samtliga ord visas i rullfönster och det gör användarens undersökning lite mer svårarbetad.

4.10.7 Användarstöd

En omfattande hjälp finns att tillgå under menyraden under Help. Här kan användaren söka efter information på flera sätt, det finns en FAQ att tillgå och till och med en hjälp om hur hjälpfilerna används. Språket är lättförståeligt och det finns många länkar inom användarstödet om användaren behöver mera information om något specifikt. Det bästa är ”steg-för-steg-modellen” över hur man kommer igång. Med i programmet finns även fyra minikorpusar som användaren kan öva sig på. Under arbetets gång kan användaren trycka F1 för att få tillgång till rätt hjälpavsnitt. R.J.C Watt skriver i sitt användarstöd att synpunkter gärna tas emot och att han vill höra ifrån dem som använder programmet. Watt vill fortsätta att utveckla sitt program i den riktning som användarna behöver.

4.11 J-Bat Kwic



4.11.1 Bakgrund

Japanen Masatoshi Sugiura vid Nagoya universitet är upphovsman till detta mycket enkla javabaserade konkordansprogram. Programmet nås via korta texter på Sugiuras hemsida. Det är för närvarande endast möjligt att göra konkordanser på de texter som ligger på denna hemsida. På hemsidan finns även en nedladdningsbar variant att tillgå.

4.11.2 Specifik funktionalitet

Detta program har endast en funktion, att skapa konkordanser.

4.11.3 Metoder och gränssnitt

Ett fönster möter användaren då hon eller han har klickat på önskad text på hemsidan. I detta fönster laddas texten och användaren får skriva in den fras, det ord (eller del av ord) som konkordansen ska behandla. Därefter finns möjligheten att välja hur många tecken som ska återfinnas till vänster och höger om den önskade strängen. När detta är gjort finns en "Search"-funktion att klicka på. Längst ner i fönstret finns

en statusrad som meddelar när sökningen är avklarad samt hur många träffar som genererades.

4.11.4 Prestanda och begränsningar

J-bat Kwic körs via webben och de enda texter som är sökbara för konkordanser är de som upphovsmannen lagt på sin hemsida, mestadels tidningstexter. Den längsta texten ligger på 1008 kB och tar runt en halv minut att ladda.¹¹ Räknare saknas, men de korta texterna är snabbgenomsökta. Programmet är känsligt för gemener och versaler och det går inte att svara för vilka alfabeterna som är möjliga då endast engelska texter finns att tillgå. Om användaren vill spara någon körning går det att markera, klippa och klistra in i det egna ordbehandlingsprogrammet. Mellan varje sökning får användaren tömma textfältet genom att klicka på funktionen Clear.

4.11.5 Test på större texter

Ett test på större texter är här inte möjligt att genomföra då programmet enbart kan nås genom att användaren klickar på en vald text som ligger på J-bat Kwics hemsida¹².

4.11.6 Resultat och omdömen

Detta program är det allra enklaste i undersökningen. Masatoshi Sugiura har skapat det i syfte att demonstrera hur ett konkordansprogram fungerar. Upphovsmannen skriver själv att programmet vänder sig till lärare som vill ha ett gratis och enkelt konkordansprogram att använda.

4.11.7 Användarstöd

Användarstöd saknas men det har ingen betydelse. Programmet är så enkelt att ett sådant stöd skulle vara överflödigt.

¹¹Texten ifråga är "Jane Eyre" av Charlotte Brontë och antalet ordformer i texten är 187.239 stycken.

¹²<http://oscar.gsid.nagoya-u.ac.jp/sugiura/calico97/KWIC/j-batkwic.html>

5 Resultatsammanställning

Alla de program som utvärderats i undersökningen är sammanställda nedan på ett grafiskt lättöverskådligt sätt. Siffrorna som visas överst i figurerna refererar till de sektioner där varje program beskrivs. Varje figur avhandlar en specifik egenskap för enkelhetens skull.

5.1 Klarar konkordansprogrammen testkorpussen?

Tabell 5.1 visar huruvida konkordansprogrammen klarar av testkorpussen som använts i denna undersökning. Ett kryss betyder ja och en ring betyder att testkorpussen inte har varit möjlig att prova på programmet ifråga. Streck betyder då att konkordansprogrammet inte klarade av testkorpussen.

Tabell 5.1: Vilka konkordansprogram klarar testkorpussen?

Program ->	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	4.10	4.11
UNT-92	X	X	X	X	O	X	-	X	-	X	O

Figur 5.1 kan vara missvisande i huvudsakligen två avseenden. 1) Testkorpussen har varit möjlig att köra, men vissa körningar har blivit så stora att programmet har hängit sig. 2) Resultatfilen blivit så stor att den varit svår att öppna. Detta gäller främst datorerna med DOS- och Windowssystem. Linux-programmen visade sig vara mer robusta i det hänseendet.

5.2 Vilka plattformar kan programmen köras på?

De flesta program är plattformsspecifika. I de fall (4.9 och 4.11) där användaren kan nå konkordansprogrammet på Internet spelar det ingen roll vilken typ av operativsystem som används. Ett program i undersökningen (4.10), som huvudsakligen är avsett för Windows, är möjligt att köra på en MacIntosh om en så kallad pc-simulator används.

Tabell 5.2: Huvudsaklig plattform för konkordansverktygen

Program ->	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	4.10	4.11
DOS			X								
Windows				X		X		X		X	
MacIntosh					X						
Linux	X	X					X				
Internet									X		X

5.3 Inom vilka språkteknologiska områden kommer programmen till nytta?

I tabell 5.3 kan språkteknologiska delområden samt konkordansprogram i undersökningen snabbasökas. Ett kryss i rutan betyder att ett språkteknologiskt delområde kan dra nytta av ett visst program. Graden av nytta är dock inte möjlig att specificera här.

Tabell 5.3: Konkordansverktygens användningsområden

Program ->	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	4.10	4.11
textanalys		X	X	X	X	X	X	X	X	X	X
morfologistudier	X	X			X			X			
kollokationsanalys			X	X		X	X	X		X	
vokabulärstudier	X	X	X	X	X	X	X	X	X	X	X
lexikografi	X			X		X	X	X		X	
korpusundersökning	X			X		X	X	X		X	

6 Diskussion

I detta kapitel summeras och diskuteras de resultat som framkommit under arbetet med undersökningen av samtliga konkordansprogram.

6.1 Konkordansprogrammets användbarhet

När en person avser att börja använda ett konkordansprogram är det några saker som hon eller han måste ta ställning till. Först och främst är arbetets art det viktiga att ta hänsyn till. Konkordansprogrammen är som tidigare noterats enbart ett verktyg för att nå ett mål och inte målet i sig. Om till exempel användaren är intresserad av statistik kring sin text och denna inte är alltför stor så är Windex ett bra val. J-Bat Kwic påminner om Windex då det också är ett nätbaserat program, men Windex har fler finesser och är alltså att föredra. Det andra en användare behöver reflektera över är vilken plattform som arbetet utförs på. Om exempelvis Windows-datorer är de som finns att tillgå så finns det fyra program i denna undersökning som kan vara lämpliga att använda sig av. Det är att rekommendera att använda sig av det system man är van vid och behärskar.

6.2 Vilka användare kan nyttja vilka program?

Personer som arbetar med textanalyser kan med fördel utnyttja samtliga program som nämns i undersökningen, dock inte Hum. Detta program upplevs som ganska begränsat och inte till så stor nytta för en textanalys. Morfologistudier kan bedrivas med fyra av undersökningens program: Hum, Konk, Conc för Mac och Kwic. Kwic är det enda av de nämnda programmen som är anpassat för Windows och har ett användarvänligt gränssnitt. Conc för Mac är också ett bra alternativ om MacIntosh är den plattform som användaren har tillgång till. Kollokationsanalyser kan utföras av sex av undersökningens program. Textine bör vara det program som är lämpligast här. Det har ett väl utformat gränssnitt och så arbetar programmet både på rad- och teckenivå. Nackdelen med programmet är att körningar tar lång tid om korpusen är alltför omfattande. IMS Corpus Workbench är det program som kan hantera störst korpus på kortast tid. Dock levereras resultatet i ett terminalfönster, men kan sparas om i en separat fil och visas i en editor. Detta kan upplevas som ett mer omständligt förfaringssätt än om resultatet visas direkt i ett programfönster. Vokabulärstudier kan utföras på samtliga program och då detta språkteknologiska delområde ofta sammanfaller med lexikografiskt arbete så visar undersökningen att störst nytta i dessa avseenden gör Hum, Textine, Conc för Mac, Monoconc, IMS Corpus Workbench, Kwic och Concordance. Även här är det IMS Corpus Workbench som klarar störst

mängd textdata. Monoconc är ett mycket brett program som dessutom ständigt förnyas. Programmet har ett användarvänligt gränssnitt och tack vare bland annat sorteringsmöjligheterna efter konkordansframtagning så är detta vara ett lämpligt program för lexikografer samt vokabulärstudier och -forskare. Den slutliga gruppen för denna undersökning är de personer som arbetar med korpusundersökningar. Sex av programmen är väl lämpade för detta arbete och bäst av dem är Kwic, Textine, Monoconc och Concordance. De är program avsedda för Windows och gränssnitten gör programmets resultat lättavlästa. Dessutom kan användaren enkelt se nyckelordens ursprung i källtexten genom enkla klick med datormusen (jämför dessa med Linux-systemens metoder där man söker med kommandorader).

6.3 Skillnader i resultat

Då jag gjorde körningarna i konkordansprogrammen tog resultaten olika lång tid att få fram, vilket var väntat då programmets kapacitet skiftar. Det som jag dock har reagerat på är vissa typer av resultat som skiljer sig från varandra, trots att svaren rimligen borde vara desamma för samtliga program. Ett exempel på ett sådant resultat är antal ordformer i korpusen. Nedan anges hur många ordformer som uppmättes i några program:

```
TSSA - 6.570.913 ordformer
Textine - 6.583.540 ordformer
Monoconc - 6.600.194 ordformer
Kwic - 6.632.428 ordformer
Concordance - 6.593.020 ordformer
```

Som tydligt kan ses ovan, så är det en stor spridning på antalet förekomster av samtliga ordformer. Skillnaden mellan största och minsta ordformsförekomst vid de utförda körningarna är hela 61.515 stycken, det vill säga omkring en procent av UNT-92-korpusens storlek. Vad som orsakar denna skillnad kan jag inte svara på. Detta kan vara en uppgift för en vidare undersökning.

6.4 Sammanfattning

Sammanfattningsvis kan sägas att dessa program som här utvärderats är ganska breda i sin utformning och de är mycket användbara för språkteknologer och andra språkintresserade. Undantaget, med avseende på bredd, är de två program som är internetbaserade, nämligen Windex och J-Bat Kwic. Då det finns många bra konkordansprogram att tillgå kan det vara mödan värt att även titta på de program som körs på andra plattformar än de man är van vid. Kanske har ett program som körs i ett annat system fler finesser som behövs för det arbete som föreligger. Om brist på tid eller intresse gör att man är förhindrad att lära in ett nytt operativsystem är det bättre att hålla sig till det system som man vanligen använder.

Det är snudd på ogörligt att precis gradera nyttan av ett specifikt program för en specifik användargrupp. Användare har självklart personliga uppfattningar om ett visst programs goda och mindre goda egenskaper. Denna utvärdering får sägas utgöra en fingervisning om vilket eller vilka program som kan vara lämpliga.

En annan viktig generell aspekt, speciellt inom forskningen, är att de program som språkteknologen eller språkforskaren ska utnyttja har en stor kapacitet och klarar av en större korpus. Ju större korpusen är, desto tillförlitligare arbetsmaterial har användaren. Om konkordansprogrammen inte behärskar större korpusar bör programmen inte användas av yrkesmänniskor till forskning.

Litteraturförteckning

- Barnbrook, Geoff. *A Practical Introduction to the Computer Analysis of Language*. Edinburgh University Press, 1996.
- Biber, Douglas. *Corpus Linguistics - Investigating Language Structure and Use*. Cambridge University Press, 1998.
- Christ, Oliver m fl. *The IMS Corpus Workbench - Corpus Query Processor: User's Manual*, 1999.
- Dahlqvist, Bengt. *TSSA 2.0 - A PC Program for Text Segmentation and Sorting*, 1997.
- Kennedy, Graeme D. *An Introduction to Corpus Linguistics*. Longman, 1998.
- Sinclair, John. *Corpus, Concordance, Collocation*. Oxford University Press, 1991.
- Smadja, Frank. Retrieving collocations from text: Xtract. *Computational Linguistics*, 1993.

A Teckenrepresentationer

Här finns länkar till de sidor på Internet där den intresserade kan få veta mer om ANSI/ASCII-koder och andra teckenrepresentationer som denna uppsats berör.

<http://czyborra.com/charsets/iso646.html>

<http://czyborra.com/charsets/iso8859.html>

<http://czyborra.com/charsets/codepages.html>

B Länksamling

Samtliga länkar som omnämns i uppsatsen finns samlade här.

<http://www.ling.lu.se/projects/Swordnet/> - Svenskt Ordnät
<http://www.swan.ac.uk/cals/calsres/index/> - engelskt vokabulärtest
<http://ariadne.ling.uu.se/bengt/windex.html> - konkordansprogrammet Windex
<http://ariadne.ling.uu.se/bengt/textin/> - Textine
<http://www.athel.com/mono.html> - Monoconc
<http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/> - IMS Corpus Workbench
<http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/misc/hum/0.html> - Hum
http://www.chs.nihon-u.ac.jp/eng_dpt/tukamoto/kwic_e.html - Kwic
<http://www.concordancesoftware.co.uk/> - Concordance
<http://oscar.gsid.nagoya-u.ac.jp/~sugiura/calico97/KWIC/j-batkwic.html> - J-bat Kwic
<http://www.sil.org/computing/conc> - Conc för Mac
<http://www.sil.org/computing/conc/tutorial.html> - handledning till Conc för Mac
<http://www.camsoftpartners.co.uk/monoconc.htm> - fler versioner av Monoconc
http://susning.nu/Regulj\%E4rt_uttryck - en sida om reguljära uttryck
<http://sv.wikipedia.org/wiki/LIX> - mer information om läsbarhetsindex

C Bigramkörning i Textine

Utdrag ur Textines bigramkörning (korpus: UNT-92) - de 200 vanligaste bigrammen.

17040 för att
14502 det är
9130 att det
7376 är det
7028 i uppsala
6793 i en
5210 är en
4872 på att
4822 kommer att
4695 av de
4597 att de
4519 att få
4503 i den
4462 är att
4397 i dag
4336 det var
4334 med en
4121 om att
3896 på en
3793 och det
3793 som är
3602 en av
3566 men det
3504 i ett
3415 med att
3343 i det
3317 det finns
3317 till att
3305 av en
3261 och en
3205 till en
3136 att man
3078 att han
3020 som en
3000 att den
2939 miljoner kronor
2873 av den

2864 för en
2854 är inte
2760 det inte
2726 i sverige
2709 en del
2659 att vi
2526 i stället
2521 i år
2521 på ett
2464 på den
2449 om det
2448 in i
2433 är ett
2338 år sedan
2319 och i
2306 för den
2250 som i
2237 när det
2225 i sin
2209 att ta
2182 det här
2112 i stockholm
2087 var det
2087 vi har
2075 som har
2050 det gäller
2044 så att
2041 del av
2029 och att
2023 den här
1993 genom att
1992 att göra
1970 tillsammans med
1949 som inte
1948 en ny
1948 inte att
1898 _det är
1893 och med
1890 för de
1886 om en
1880 fram till
1880 och den
1870 i de
1822 vad som
1821 med ett
1819 och de
1802 ett par
1794 det som
1788 att en

1762 att ha
1751 har en
1722 det blir
1720 till den
1708 är den
1681 med den
1680 även om
1675 de flesta
1672 sig i
1662 blir det
1647 att bli
1620 av det
1599 mer än
1594 han är
1590 på det
1589 den som
1584 har varit
1581 den nya
1580 av att
1569 det kan
1568 procent av
1561 har inte
1507 att vara
1499 stockholm tt
1493 en stor
1490 till ett
1485 om de
1476 en annan
1475 för det
1463 var en
1462 det har
1461 kan man
1455 om man
1439 för ett
1417 har vi
1417 nästa år
1409 svårt att
1398 om den
1393 inte bara
1386 är i
1357 och har
1355 när han
1339 dem som
1333 inte är
1332 som kan
1330 den svenska
1324 bland annat
1321 som ett
1316 jag har

1309 i tierp
1308 man kan
1296 man inte
1295 sig till
1279 och ps
1278 också att
1272 de som
1268 har det
1268 har man
1266 _ det
1262 finns det
1259 år fyller
1257 av ps
1257 som nu
1256 av dem
1256 utan att
1251 samband med
1250 nu är
1247 säger han
1244 det att
1243 han har
1232 och som
1221 han var
1211 sig att
1208 och ett
1207 av ett
1202 i samband
1196 trots att
1194 har också
1194 med i
1193 att kunna
1191 officiant var
1184 till de
1180 född i
1177 vid en
1175 i enköping
1167 det i
1167 som talade
1166 på de
1149 i början
1149 talade över
1145 att jag
1143 som det
1134 och andra
1126 till det
1116 ett av
1111 sig på
1109 har fått
1109 har han

1102 vi inte
1094 på fredagen
1087 som de
1087 är mycket
1085 att se
1080 när de
1077 hölls på
1072 innebär att
1072 med de
1071 i landet
1069 så mycket
1063 som avslutning
1059 att gå
1052 efter en
1043 som den
1042 på tisdagen

D Loggfil i TSSA

Här är ett exempel på hur en loggfil kan se ut i TSSA. (korpus: Markurells i Wadköping)

```
TTTTTTTTTT      SSSSSSSS      SSSSSSSS      AAAAAAAA
TTTTTTTTTT      SSSSSSSSSS     SSSSSSSSSS     AA      AA
  TT            SS      S      SS      S      AA      AA
  TT            SS            SS            AA      AA
  TT            SSSSSSSS     SSSSSSSS     AA      AA
  TT            SSSSSSSS     SSSSSSSS     AAAAAAAAAA
  TT            SS            SS            AAAAAAAAAA
  TT            S      SS     S      SS     AA      AA
  TT            SSSSSSSSSS     SSSSSSSSSS     AA      AA
  TT            SSSSSSSS     SSSSSSSS     AA      AA
```

TSSA VERSION 2.1

DATE: 105-05-20

TSSA COPYRIGHT (c) 1995, DEPT. OF LINGUISTICS, UPPSALA UNIVERSITY

Input file: MARKUREL.TXT

Reserved chars:

Upper case mark = \ Section mark = # Phrase mark = +

Subtext markers = < > Wildcard mark = *

Total no of words in text: 37747

Total no of characters in text: 231524 (including blanks)

11 word delimiters defined: .,:;?!"()/

No of predefined characters: 92 (from file tssa.inf)

Type: 6805 Token: 37747

Type-token ratio: 0.180

LENGTH DISTRIBUTION OF WORDS IN TEXT:

Length	Freq	Percent	
1	868	2.30	***
2	3986	10.56	*****
3	11250	29.80	*****
4	5254	13.92	*****
5	4232	11.21	*****
6	3697	9.79	*****
7	2490	6.60	*****
8	1739	4.61	*****
9	1798	4.76	*****
10	818	2.17	***
11	599	1.59	**
12	434	1.15	*
13	236	0.63	*
14	121	0.32	
15	157	0.42	*
16	37	0.10	
17	12	0.03	
18	11	0.03	
19	5	0.01	
20	3	0.01	

Word length, mean: 4.84 stdev: 2.68 range: 19

FREQUENCY DISTRIBUTION OF PREDEFINED CHARACTERS IN TEXT:

No	Char	Frq	No	Char	Frq	No	Char	Frq	No	Char	Frq
1	A	96	17	h	5496	33	p	2705	49	y	1034
2	a	16773	18	I	98	34	q	1	50	z	7
3	B	148	19	i	8056	35	R	162	51	z	4
4	b	1938	20	J	424	36	r	14361	52	"	8
5	C	117	21	j	1549	37	S	268	53	0	12
6	c	2914	22	K	128	38	s	10440	54	1	40
7	D	542	23	k	5995	39	T	193	55	2	9
8	d	9095	24	L	292	40	t	14437	56	3	15
9	E	135	25	l	9107	41	U	41	57	4	4
10	?	4	26	M	915	42	u	4185	58	5	5
11	e	17728	27	m	5088	43	V	220	59	6	15
12	F	230	28	N	120	44	W	112	60	7	2
13	f	2855	29	n	15418	45	v	3792	61	8	3
14	G	58	30	O	297	46	w	37	62	9	21
15	g	6930	31	o	7061	47	x	108			
16	H	860	32	P	92	48	Y	1			

Total no of predefined characters in text: 172801

FREQUENCY DISTRIBUTION OF UNDEFINED CHARACTERS IN TEXT:

No	Char	Frq	No	Char	Frq	No	Char	Frq	No	Char	Frq
63	ö	3156	67	é	21	71	Å	28	75]	7
64	ä	3578	68	Ä	33	72	§	1	76	æ	4
65	å	2819	69	ü	119	73	¨	1			
66	-	3870	70	Ö	34	74	[7			

Total no of undefined characters in text: 13678

E Trigramkörning i TSSA

Ett trigram baserat på UNT-92-korpusen, körd i TSSA - de tjugo mest frekventa trigrammen.

1189 i samband med
1139 som talade över
1068 att det är
1044 en av de
999 och det är
985 som avslutning spelades
939 för att få
913 på grund av
902 akten omramades av
892 omramades av ps
861 en del av
844 att det inte
794 år fyller på
754 kyrkogård minnesstund hölls
742 i slutet av
736 blommor märktes från
722 minnesstund hölls i
666 talade över orden
663 över orden i
640 men det är