Institut für Computerlinguistik
Universität Zürich

Sommersemester 2003

Seminar „Korpuslinguistik für und mit Computerlinguistik“
lic. phil. Gerold Schneider
lic. phil. Simon Clematide

Programm zur Bestimmung von schwedischen Nebensätzen am Beispiel der att-Nebensätze mit Ausblick auf om- und som-Nebensätze

Therese Rutishauser
Mattastr. 38
7270 Davos Platz
thrutishauser@hotmail.com

Abgabedatum: 26.5.2003

Fragestellung

Wie weit lässt sich die Suche nach schwedischen Nebensätzen aus früheren Jahrhunderten automatisieren? Sie müsste sich an zeit- und autorenspezifische Orthographie und syntaktische Eigenheiten anpassen lassen. Im Folgenden wird ein Programm vorgestellt, das wichtige Nebensatzanfänge identifiziert und damit dem Text eine erste Struktur gibt. Da nicht davon ausgegangen werden kann, dass ältere Texte mit Wortartentags versehen sind, werden im Programm direkt Zeichenketten (Wortformen und Satzzeichen) gematcht. Diese Zeichenketten können einfach an die zeit- und autorenspezifische Orthographie angepasst werden.
Zum Schluss stellt sich die Frage nach der Effizienz des Programms. Es findet häufige Nebensatzanfänge im gesamten Text, ist aber als erster Schritt bei der Analyse von Verbformen im Innern von Nebensätzen vorgesehen. In einem Experiment wird getestet, ob direkt in der Umgebung von bestimmten Verbformen strukturelle Merkmale von Nebensätzen gesucht werden könnten. Dabei wird festgestellt, dass eine Umgebung von bis zu 60 Zeichen, die nicht von Satzzeichen unterbrochen wird, gute Resultate liefert.

INHALT

Abkürzungen

A	Satzadverbiale (Satzstruktur)
AKK	Akkusativ
ART	Artikel
DEF	bestimmter suffigierter Artikel
F	Finite Verbform (Satzstruktur)
FIN	Finite Verbform (im Programm)
IMPF	sog. Imperfektform
INF	Infinitiv
NOM	Nominativ
NS	Nebensatz
PART	Partizip
PL	Plural
PREP	Präposition oder Präfix
SAG	einige Verben des Sagens
SPRECH	einige Verben des Sprechens
SUP	Supinum
WIE	Vergleich ohne Nebensatz

Definitionen

Subjunktion: Hypotaktische oder subordinierende Konjunktion.
Konjunktion: Parataktische oder koordinierende Konjunktion.
Teilsatz: Haupt- oder Nebensatz; enthält typischerweise eine finite Verbform.

1. Frei zugängliches Korpus: Spraakbanken

Die Sammlung von Texten aus 'älteren schwedischen Romanen' kommt zeitlich den zu untersuchenden Texten am nächsten. In ihr kann nach Zeichenketten gesucht werden und "seitenweise" (ca. 30 Druckseiten) Originaltext heruntergeladen werden. Zudem enthält die Sammlung die 1000 häufigsten Wortformen mit ihren absoluten Häufigkeiten. Alle drei Teile werden im Folgenden benutzt, um ein Programm zu entwickeln, das mit geringem Aufwand möglichst viele Nebensätze identifizieren kann.

2. Struktur typischer schwedischer Nebensätze

In typischen schwedischen Nebensätzen steht die Satzadverbiale (A) vor der finiten Verbform (F). Diese Teilsätze werden in der schwedischen Grammatik als af-Sätze bezeichnet (SAG I:150).

2.a) Beispiel: Hauptsatz mit fa-Struktur gefolgt von zwei untergeordneten Teilsätzen

F A (1a) Du vet nog, Du weisst wohl,
F (1b) att vi har funn-it sten-ar på strand-en, dass wirNOM haben finden-SUP Stein-PL auf Strand-DEF,
A F (1c) som inte har var-it anna-t än förhårdn-ade mask-ar.[1] die nicht haben sein-SUP anders-DEF als hartwerden-PART Wurm-PL.
Du weisst wohl, dass wir am Strand Steine gefunden haben, die nichts anderes waren als vertrocknete Würmer.

Nebensätze unterscheiden sich strukturell nur dann von Hauptsätzen, wenn sie eine Satzadverbiale enthalten. Das heisst, dass zum Beispiel (1b) nicht aufgrund der Wortstellung, sondern nur aufgrund der Einleitung 'att' als Nebensatz identifiziert werden kann. Im Gegensatz dazu können im Deutschen typische Nebensätze meist mit Hilfe der Stellung des finiten Verbs eindeutig von Hauptsätzen unterschieden werden.

2.b) Resultat der Korpusabfrage "nicht hat(te)"

Sucht man in dieser Weise nach "häufiger Adverbiale vor häufigem Finitum" so findet man 248 "inte har" und 526 "inte hade" in 'äldre svenska romaner' von Spraakbanken (60 Zeichen Umgebung). Davon sind drei von 40 genauer untersuchten "inte har" eigentlich "Inte har", das heisst ein emphatischer Hauptsatzanfang, und eins von 20 untersuchten "inte hade" ist eigentlich ein "inte, hade". Somit stehen mehr als 90% der in der Konkordanzsuche von Spraakbanken gefundenen Textstellen in Nebensätzen. Die häufigsten Subjunktionen sind hier 'att', 'om' und 'som'.
Da mit dieser Abfrage weder Nebensätze ohne Adverbiale noch vollständige Nebensätze gefunden werden, wird im Folgenden ein Programm entwickelt, das die häufigsten Nebensatztypen auf ca. 30 Seiten aus Selma Lagerlöfs 'Nils Holgerson' erkennen soll.

3. Programm zur Bestimmung von Teilsatzanfängen

3.a) Vorüberlegungen: Die häufigsten Nebensatzanfänge

Wortformen aus 'die 1000 häufigsten' in 'äldre svenska romaner' nach möglichen Wortarten und absoluter Häufigkeit sortiert:

mögl. Subjunktionen. Problem: Wörter, nicht Funktionen -> ambig	att, som, om, då, när, vilken (dass, Relpron, ob, als, wenn, welcher)
(mögl.) Konjunktionen	och, eller, ty (und, oder, denn)
Finite Verbformen	var, är, hade, har, skulle, sade,
Pronomina eindeutig im Nominativ	han, jag, hon, du, vi, ni
Supinum	varit, blivit, fått, kommit, sett, gjort, gått

[1] Genauer Fundbeleg in Spraakbanken:" sten. Du vet nog, att vi har funnit stenar på stranden, som inte har varit annat än förhårdnade maskar. Och nu tror jag, lage_nils Sida Kon [Söksträng: inte har varit] [Material: romg] [Fraser: 14], [Ingångsidan] | [Upp]. Sökning utförd kl. 7.20 den 30 april 2003"

In Übereinstimmung mit den Beobachtungen in 2.b) ist anzunehmen, dass 'att'-, 'som'- und 'om'-Nebensätze am häufigsten auftreten.

3.b) Programmierschritte

Der zu analysierende Text enthält nur am Ende von Abschnitten Zeilenwechsel und dort jeweils einen doppelten. Um die Struktur sichtbar zu machen soll jeweils am Satz- und später auch Teilsatzende ein Zeilenwechsel eingefügt werden. In einem ersten Schritt werden die Satzendezeichen speziell markiert und durch einen Leerschlag von vorangehenden Wort getrennt. Im zweiten Schritt werden alle " att ", " om " und " som " durch " NS_att ", " NS_om " bzw. " NS_som " ersetzt. Darauf wird bei allen Satzendezeichen und " NS_" ein Zeilenwechsel eingefügt. Nachdem das soweit entwickelte Programm auf den zu analysierenden Text angewendet worden war, konnte im Output die Einwortumgebung von 'att' genauer untersucht werden. Dabei zeigte sich, dass sich die Umgebung von 'att'(='dass') und 'att'(='zu') folgendermassen unterscheidet:

'att' (= dass), d.h. Subjunktion	'att' (= zu) vor Infinitiv
nachher: ein Pronomen im Nominativ anpassbare Liste möglich	nachher: ein unregelmässig gebildeter Infinitiv anpassbare Liste möglich
vorher: ein Komma	Infinitivendung im Normalfall '-a'
nachher: irgendeine Nominalphrase	vorher: ein vom vorangehenden Verb abgetrenntes Präfix

Aufgrund dieser Beobachtungen wurde das Programm in einem dritten Schritt so ausgebaut, dass es zwischen nebensatzeinleitendem 'att' (='dass') und 'att' (='zu') vor einem Infinitiv unterscheiden kann.

Die unten vorgestellten Programmzeilen führen dazu, dass in jeder Zeile des Originaltexts Zeichenketten so oft wie möglich (g: global) durch andere Zeichenketten ersetzt (s: substituiert) werden.

1. Satzendepunkte

```
s/\.\s/ PUNKT\.PUNKT /g;
#Hier vorerst nur Punkt vom letzten Wort wegrücken und kennzeichnen.
```

2. 'att', 'som', 'om' --> 'Nebensatzeinteilung'

```
s/( Om | om )/ NS_om /g;
#mögliche NS-Anfänge 'att', 'som', 'om' analog markieren.
```

3. 'att' grob desambiguiert

```
#'att' Subjunktion 'dass' oder Infinitivmarkierung 'zu'?
#zuerst nur möglichst sichere Fälle kennzeichnen mittels 'matchen' von
#Zeichenketten (Wortformen, Leerzeichen...)
s/( Att | att )(han|hon|jag|det|den|de|någon) / NS_att $2 /g;
                    #Häufige Pronomina am Anfang der 'att'-Nebensätze
s/, att /, NS_att /g;   #Komma vor 'att' wichtiger Hinweis für NS-Einl.
s/( på att )/ på INF_att /g;
s/( Att | att )(((\w|ä|ö|å)+)a) / INF_att $2 /g;
s/( Att | att )(inte|se|bli|stå|få|fly|gå|dö|tro) / INF_att $2 /g;
                    #Infinitive direkt nach 'att' erkennen:
                    #Endung auf -a und Ausnahmeliste.
                    #PROBLEM: Fehler, wenn nach 'att' ein Wort
                    #steht, das auf -a endet und kein Infinitiv ist.
                    #Bsp. 'hela Västerbotten' (='ganz Västerbotten')
 s/( Att | att )/ NS_att /g;
                    #'att', die nicht bereits identifiziert wurden,
```

```
#werden hier als 'NS_att' klassifiziert.
```

Resultat der att-Desambiguierung mit diesem Programm:
Erste drei Seiten[2]: Alle neun INF_att und 13 NS_att werden richtig erkannt.
Letzte drei ganze Seiten: Acht INF_att und 28 NS_att werden richtig erkannt. Aber fünf von 41 Entscheidungen sind falsch: ein INF_att nämlich 'begravas' (='begraben werden') wird als NS_att klassifiziert, weil er auf das Passivsuffix '-s' endet. Dieser Fehler ist nicht unbedingt behebbar, da auch andere Wörter auf '-s' enden.
Vier INF_att werden als NS_att klassifiziert, da nach ihnen kein Leerschlag sondern ein Komma steht. Diese häufigen Fehler können behoben werden, indem wie bereits vor Punkten auch vor Kommas, Strich- und Doppelpunkten ein Leerschlag eingefügt wird. Nach der Korrektur ist noch eine der 41 Entscheidungen falsch. Im Programmtext wird ausserdem 'förstå' (='verstehen') in die Liste der unregelmässigen Infinitive aufgenommen. Zudem werden zwei linguistisch sinnvollere Zeilen für das Auffinden von 'att inte INFINITIV' ins Programm eingefügt und dafür 'inte' (='nicht') aus der Liste der oft auf 'att' (='zu') folgenden Wörter entfernt, die jetzt nur noch unregelmässige Infinitive enthält.

Auswertung[3] (Programm in Anhang A)

NS_att insgesamt 222		INF_att insgesamt in 110 Zeilen[4]
att han (er)	30	INF(Ausnahmeliste) in 33 Zeilen
att hon (sie)	34	INF(Endung) in 78 Zeilen Davon 77 richtig[5] und eine falsch identifiziert. Fehler bei: 'så att mera än begravningar...' (= 'so dass mehr als die Begräbnisse...') Grund: 'mera'(='mehr') endet auf '-a'.
att jag (ich)	4	
att det (es,…)	40	
att den (es, ...)	6	
att de (die,…)	26	
att någon (jemand)	1	
, att	72 alle in Ordnung	
'DEFAULT' att	9 8 richtig und 1 falsch (Infinitiv Passiv 'begravas')	

Im Ganzen wird also bei diesem Text, der etwas mehr als 330 'att' enthält, nur ein 'dass' als 'zu' und ein 'zu' als 'dass' qualifiziert. Allerdings werden verhältnismässig viele Entscheidungen aufgrund des Kommas vor dem 'att' gefällt.

4. Fazit zur Desambiguierung der att-Sätze

Wäre die Kommasetzung weniger konsequent durchgeführt, könnte eine Liste mit häufigen Nichtinfinitiven auf '-a' erstellt werden und weiter unten im Programm neben 'inte' auch 'bara' (='nur') zwischen 'att' und einem Infinitiv erlaubt werden. Zudem könnte die

[2] von 32 Seiten Output zum in 2.b) definierten Lagerlöftext
[3] In dieser und in den folgenden Auswertungen werden nur absolute Häufigkeiten angegeben, da die Unterteilung in Einzelprobleme sehr kleine Zahlen mit sich führt und genau diese Einzelprobleme bei der Anpassung des Programms an andere Texte benutzt werden können. Diese Auswertung bezieht sich auf den in Abschnitt 2.b) erwähnten Lagerlöftext.
[4] 110 entspricht nicht genau der Anzahl der 'att', die vor einem Infinitiv stehen, weil in einzelnen Zeilen mehr als einmal 'att' vor einem Infinitiv vorkommt.
[5] An einer Stelle allerdings mit Hilfe des Zufalls: 'att bara räfsa gården' (='nur den Hof zu rechen'). Das Wort 'bara' (='nur') zwischen 'att' und dem Infinitiv 'räfsa' (='rechen') hat zufälligerweise die Endung '-a'.

'Passivinfinitivendung' '-s' vermutlich mit zufriedenstellender Treffsicherheit identifiziert werden.[6] Ausserdem kann die Liste der Pronomina und jene der unregelmässigen Infinitivformen erweitert werden. Somit ist das Ziel erreicht: Das Programm identifiziert att-Nebensatzanfänge zuverlässig und lässt sich einfach an die orthographischen Begebenheiten älterer Texte anpassen.
Im Abschnitt 5.b) und 5.c) wird versucht auch die om- bzw. som- Nebensatzanfänge zu identifizieren, da 'som' und 'om' ebenfalls sehr häufig in schwedischen Texten auftreten, aber bei weitem nicht immer einen Nebensatz einleiten.
Mit dem gewählten Ansatz, bei dem Nebensatzanfänge identifiziert werden, wird es nicht möglich sein Nebensätze als Ganzes zu erkennen, da nicht einmal das Ende eines eingebetteten Nebensatzes bestimmt wird.
Die att-Desambiguierung ist überraschend zuverlässig. Das dürfte auch daran liegen, dass im untersuchten Lagerlöftext verschiedene vereinfachende Bedingungen vorliegen. Zum einen sind in Spraakbankens 'äldre svenska romaner': Interpunktion und Leerschläge sorgfältig aufbereitet. Zum andern ist die Orthographie der Romane aus dem 19. Jahrhundert nicht sehr veraltet. Diese Romane enthalten zum grössten Teil vollständige wohlgeformte Sätze und praktisch keine Abkürzungen, Zahlen, Daten etc. mit Punkten. Das bedeutet, dass Punkte fast immer ein Satzende markieren. Das Programm an die wirklichen Bedingungen in älteren Texten anzupassen dürfte einiges aufwändiger sein, als das Ergänzen von z.B. Infinitivformen.

5. Ausbau des Programms

5.a) att-Sätze

Der Teil zur Erkennung der att-Sätze kann für den in 2.b) ausgewählten Testtext nicht mehr verbessert werden. Dennoch werden im Code einige Muster hinzugefügt: 'att' vor einem unbestimmten Artikel 'en' oder 'ett' oder einem Nomen mit suffigiertem, bestimmtem Artikel ('-n', '-t') leitet einen Nebensatz ein. Die 'Infinitivpassivendung' '-s' wird ebenfalls provisorisch zur Infinitiverkennung eingesetzt. Damit wird die att-Desambiguierung weniger abhängig von der Kommasetzung. Allerdings ist das Komma bis hierhin immer noch das einzige Indiz mit dem Adjektive auf '-a', wie sie in unbestimmten Pluralnominalphrasen[7] üblich sind, von typischen Infinitivformen auf '-a' unterschieden werden können. Das ändert sich, wenn man die betrachtete Umgebung von 'att' auf mehr als ein Wort ausdehnt. Folgt nach 'att' ein einziges Wort und dann gleich ein Satzendezeichen (. ? !), so muss es sich um ein 'zu' handeln, da typische schwedische Nebensätze mindestens zwei Worte nämlich ein Subjekt und ein Verb enthalten müssen. Ein 'att' kurz nach 'sagen', 'berichten' und 'träumen' muss ein 'dass' sein. 'Kurz' wurde implementiert als Abstand von Null bis zwei Wörtern und eventuell einem Komma zwischen dem Verb und 'dass'.
Zum Schluss werden nebensatzeinleitende 'dass' in Kombinationen wie '', so dass '' oder '', ohne dass '' bestimmt.

[6] Ich habe im hier untersuchten Lagerlöftext kein anderes auf '-s' endendes Wort gefunden, das direkt hinter 'att' steht.
[7] Bsp. 'stora hus' (='grosse Häuser') kann ohne Artikel direkt auf 'att' (='dass') folgen.

5.b) om-Sätze

Nebensatz 'ob', bzw. 'wenn…nicht'	Präposition 'über' bzw. 'in' (temporal)
(2a) Jag vet inte, Ich weiss nicht, om han (ha-r) köp-t bröd. ob er (ha-t) kauf-SUP Brot.	(2b) Det ska vi tal-a om. Das werden wir sprech-INF über. (2c) Dörr-en öppn-ade-s om natt-en. Tür-DEF öffn-IMPF-PASSIV in Nacht-DEF

Sowohl nach einem nebensatzeinleitenden 'om' als auch nach der Präposition 'om' steht in der Regel eine Nominalphrase. Somit ist eine Desambiguierung schwieriger als bei den att-Sätzen.
Im Programmcode werden der Reihe nach folgende Merkmale zur Unterscheidung von nebensatzeinleitenden und nicht nebensatzeinleitenden 'om' implementiert:
Da schwedische Personalpronomen (ausser in der dritten Person Plural) unterschiedliche Nominativ- und Akkusativformen haben und ein nebensatzeinleitendes 'om' Nominativ verlangt, während auf die Präposition 'om' Akkusativ folgt, kann im Fall von Personalpronomen nach 'om' desambiguiert werden.
Weil ein Nebensatz mindestens ein Subjekt und ein Verb enthalten muss, kann er nicht aus nur einem Wort bestehen.
Folgt auf ein 'om' direkt ein Teilsatzende (Satzzeichen) bzw. ein Teilsatzanfang, handelt es sich bei 'om' um eine Präposition, die sich wie in (2b) auf eine weiter vorne stehende Nominalphrase bezieht.
'Om' vor einem temporalen Ausdruck wie 'natt-en' (='Nacht'-DEF) oder 'två timm-ar' (='zwei Stunde'-PL) kann meist mit der temporalen deutschen Präposition 'in' wiedergegeben werden.
Im Weiteren ist ein Komma vor 'om' ein Hinweis auf einen Nebensatzanfang.
Nach einigen Verben des Sprechens entspricht 'om' der deutschen Präposition 'über'. Implementiert wurde: "Bis vier Wörter nach diesem Verb steht 'om'".
In den nächsten beiden Klauseln wird verlangt, dass mindestens im Abstand von einem Wort nach dem 'om' aber vor dem nächsten Satzzeichen eine finite Verbform steht. Die erste dieser beiden Klauseln erkennt einige häufige Verbformen, die zweite Endungen auf '-de' und '-it'. Dabei ist '-de' die reguläre Imperfektendung und '-it' die Endung einiger typischer Supinumsformen. Diese werden hier gesucht, da vor Supinumsformen in Nebensätzen das finite Hilfsverb fehlen kann. Nach Präsensformen wird nicht gesucht, weil sich ihre Endung '-r' nicht von der regulären Pluralendung von Nomen unterscheidet.
Die verbleibenden 'om' werden defaultmässig gekennzeichnet.

Auswertung (Programmtext in Anhang B)

NS insgesamt 36 identifiziert		PREP insgesamt 33 identifiziert Zusätzlich: 8 der 10 DEFAULT-Entscheide keine Nebensätze	
NOM	14	AKK	6
		LAENGE	7
		POSITION	1
KOMMA	12	TEMP	4
FIN	11	SPRECH	15
		DEFAULT	10 davon 2 NS aber 8 keine NS

Mit Ausnahme der Defaultentscheidungen identifiziert das Programm im getesteten Lagerlöftext alle 'om' richtig. Da ich auch ungewöhnlich konstruierte Nebensätze, wie den einen der beiden Defaultfälle, der einen eingebetteten Relativsatz enthält, finden möchte, markiere ich die Defaultentscheidungen, um sie noch von Hand auswerten zu können.

5.c) som-Sätze

Relativsatz 'der', 'die', 'das', 'dem',… seltener 'wie' als Nebensatzeinleitung.	Vergleich 'wie ohne Nebensatzeinleitung'
(3a) sten-ar på strand-en, som inte har var-it anna-t än förhårdn-ade mask-ar.[8] Stein-PL auf Strand-DEF, die nicht haben sein-SUP Anderes-DEF als verhärte-te Wurm-PL. (3b) Som du har sag-t, regnar det. Wie du hast sag-SUP, regnet es.	(3c) Han är lika stor som du. Er ist gleich gross wie duNOM.

Zur Unterscheidung von Relativsätzen und Vergleichen kann weder Kasus, noch Teilsatzlänge herangezogen werden.
Im Programm werden die folgenden beiden Hinweise auf Nebensätze benutzt: Entweder es steht eine eindeutig finite Verbform nach 'som' und vor dem nächsten Satzzeichen oder unmittelbar vor 'som' hat es ein Komma.
Der Hinweis auf 'wie ohne Nebensatzeinleitung', der im Programm benutzt wird, ist: 'samma', 'lika', 'så', (='gleich', 'gleich', 'so') steht im Abstand von bis zu vier Wörtern vor 'wie'.
Die verbleibenden 'som' werden wiederum defaultmässig separat gekennzeichnet, damit sie bei der Anwendung des Programms auf andere Texte einfach überprüft und für eine Programmverbesserung verwendet werden können.

Auswertung (Programmtext in Anhang B)

NS insgesamt 164 bestimmt		wie-Vergleich ohne Nebensatz 11 korrekt identifiziert 15 DEFAULT Fälle	
FIN[9]	120	WIE0	0
KOMMA[10]	44 davon 1 falsch und 43 richtig	WIE1	9
		WIE2	0
		WIE3	0
		WIE4	2
		DEFAULT	15 Alle 15 keine Nebensätze.

[8] Ausschnitt aus (1) in Abschnitt 2.a).
[9] Alles Nebensätze. Einer davon ("Det var en vildgås , NS_FINsom låg och sov i ett litet rede på marken , och bredvid henne stod gåskarlen PUNKT.") wird mit Glück als Nebensatz erkannt, weil 'rede' (='Nest') vom Programm als finite Verbform angesehen wird. Die beiden starken Präteritumsformen 'låg' (='lag') und 'sov' (='schlief') kennt es nämlich nicht. Das Kommakriterium würde diesen Nebensatz auch finden.
[10] Der Fehler ist das zweite NS_KOMMAsom in: "Han kände inte igen den runda sjön , NS_KOMMAsom låg mitt i dalen , och aldrig någonsin hade han sett sådana eländiga , förkrympta björkar , NS_KOMMAsom de han nu låg under PUNKT." (=…Birken, wie die [wörtlich: er nun lag unter] unter denen er nun lag).

Aufgrund einiger häufiger finiter Verbformen, der regulären Vergangenheitsformen, der Kommata und an einer in Fussnote neun erwähnten Stelle mit Hilfe von Zufall werden alle som-Nebensätze im Text gefunden. Ein 'som' wird aufgrund eines Kommas fälschlicherweise als nebensatzeinleitend identifiziert (vgl. Fussnote 10). Es wurde geprüft, ob die ebenfalls häufige Präteritumsendung '-te' benutzt werden könnte, um die Abhängigkeit des Programms von Kommas zu verringern. Aber es enden zu viele andere Wörter auf '-te'.[11]
Alle 11 vom Programm gefällten Entscheide zu WIE sind richtig. 'gleich' im Abstand 0 vor WIE kommt nicht vor. Diese Klausel könnte wohl aus dem Programm entfernt werden, denn bei Vergleichen wie 'gleich gross wie…' und 'so lang wie…' muss typischerweise mindestens ein Wort zwischen 'gleich' oder 'so' und 'wie' stehen.

6. Test: att-Sätze in einem Text mit spezieller Orthographie

Die Behauptung in 4. Fazit: "Das Programm identifiziert att-Nebensatzanfänge zuverlässig und lässt sich einfach an die orthographischen Begebenheiten älterer Texte anpassen" soll hier an einem Text getestet werden, der bei der Korpusabfrage in Spraakbankens 'ältere schwedische Romane' durch ein fehlendes Komma und die Verwendung der Form 'ha' als finite Pluralform von 'haben' aufgefallen ist:

(4a) Och högfärdig-a ha de alltid var-it så Und hochmütig-PL habenPL sie immer sein-SUP so
(4b) att en torpardotter inte var-it människa för dem.[12] dass eine Bettlertochter nicht sein-SUP Mensch für sie.

Ein Test der in 5.a) verfeinerten Version zur att-Desambiguierung zeigte, dass das Programm, das finite Pluralformen auf '-a' nicht von Infinitiven auf '-a' unterscheiden kann, einige Fehler machte. Vier dieser Fehler konnten vermieden werden, indem wie bei om-Sätzen ab dem zweiten Wort nach der Subjunktion und vor dem nächsten Satzzeichen finite Verbformen gesucht wurden.

[11] Im untersuchten Text gibt es 259 Endungen auf '-te', 148 davon betreffen das Wort 'inte' (='nicht'), zudem ist '-te' auch Superlativendung.
[12] Gemäss Spraakbanken aus: Bergman, Hjalmar, Mor i Sutre 1899. (In Spraakbanken leider ohne Seitenangaben)

Auswertung der nun vorliegenden Version zur att-Desambiguierung[13] (Anhang B)

NS_att insgesamt 70 (67 richtig, 3 falsch)		INF_att insgesamt 26 (alle 26 i.O.)
att han (er)	9	INF(Ausnahmeliste) 2
att hon (sie)	3	INF(Satzlänge zu kurz für NS) 4
att jag (ich)	9	
att det (es,…)	7	
att den (es,…)	0	
att de (die,…)	6	INF(Endung) 20 Alle 20 richtig[14], auch 'tas' (='genommen werden') erkannt; 'glas' (='Glas') wurde in FIN erfolgreich abgefangen.
att någon (jemand)	1	
att (du| vi| ni)	2	
att ART	19 18 i.O.[15] 1 falsch[16]	
SAG att	0	
, att	2	
att FIN	7 6 i.O.[17] 1 falsch[18]	
att '-e'	1 1 falsch[19]	
'DEFAULT' att	4 4 richtig	

Die Resultate zeigen, dass das Programm auch bei diesem Text alle att-Nebensätze findet und nur drei infinitivmarkierende 'att' als nebensatzeinleitend identifiziert, nachdem es an einer Stelle angepasst wurde.
Die Identifikation von att-Nebensätzen erreicht also, wie in Abschnitt vier vermutet, das Ziel: Sie lässt sich einfach an andere Texte anpassen. Allerdings müssen die Programmteile zu den übrigen Subjunktionen separat angepasst werden. Zudem wäre es wünschenswert, dass das Programm auch weniger häufige Subjunktionen erkennen könnte.

7. Könnte auf die Bestimmung von Teilsatzgrenzen verzichtet werden?

Zum Schluss stellt sich die Frage nach der Brauchbarkeit des Programms. Es ist als erster Schritt bei der Analyse von Verbformen[20] im Innern von Nebensätzen vorgesehen. In einem Experiment wird getestet, ob direkt festgestellt werden könnte, ob in der Umgebung einer Supinumsform das finite Hilfsverb 'haben' "fehlt". Das ist heute in typischen schwedischen Nebensätzen möglich.

7.a) Resultat eines Experiments

Die Konkordanzanfrage "inte varit" (='nicht gewesen', 'varit' als Beispiel eines häufigen Supinums) an Spraakbanken ('äldre svenska romaner', Umgebung: vorn und hinten 60 Zeichen) liefert 158 Textstellen aus vielen Romanen von verschiedenen Autoren. Dabei ist anzunehmen,

[13] Programmtext in Anhang B.
[14] An einer Stelle allerdings nur mit Glück: 'att bara räfsa gården' (= 'nur den Hof zu rechen'). Das Wort 'bara' (='nur') zwischen 'att' und dem Infinitiv 'räfsa' (='rechen') hat zufälligerweise die Endung '-a'.
[15] Glück bei 'man' und 'länsman': '-n' ist kein suffigierter Artikel, aber 'man' (='Mann') ist auch Infinitiv.
[16] "Att jämt gå…" die Satzadverbiale 'jämt' (='immerzu') hat die Endung '-t'.
[17] Bei vier Substantiven und Adjektiven auf '-a' und '-as' wird eine Klassifizierung als Infinitiv verhindert.
[18] Da kein Komma vorhanden ist, wird eine finite Verbform in einem untergeordneten Teilsatz einem 'att + Infinitiv' zugeordnet.
[19] In: "att bättre kunna" (='besser zu können') steht 'bättre' (='besser') vor einem Infinitiv.
[20] Speziell: Supinum mit oder ohne finites Hilfsverb 'haben' (Rutishauser: 28-31).

dass "inte varit" in vielen Fällen in Hauptsätzen steht, in denen das flektierte 'haben' vor 'inte' steht. Deshalb werden in verschiedenen Schritten Sätze mit finitem 'haben' direkt vor, bzw. an zweiter oder dritter Stelle vor 'inte varit' gestrichen. Dabei tritt keine falsche Streichung auf. Das heisst es werden keine Teilsatzgrenzen so weit überschritten, dass ein flektiertes 'haben' im übergeordneten Teilsatz die Streichung verursacht. Dagegen enthielten noch fünf von 42 Textstellen 'har' bzw. 'hade' an bis zu siebter Stelle vor 'inte' und hatten fa- Struktur, das heisst Hauptsatzstruktur.
Die sechs Textstellen mit Hauptsatzstruktur, in denen ein flektiertes Modalverb ('skulle', 'måtte'), vor 'inte varit' steht, könnten problemlos ausgefiltert werden. Ein Problem bereitet die an zwei Stellen nicht erkannte ältere finite Verbform 'ha' (='haben'). Sie könnte autorenabhängig ins Filterprogramm aufgenommen werden, ist aber ambig, da der Infinitiv die gleiche Form haben kann.

7.b) Verbesserung des Filterprogramms

Das Programm kann vielleicht verbessert werden, indem aus der Liste der Konkordanzen von 'inte varit' diejenigen Teilsätze als 'Hauptsätze' identifiziert, gespeichert und gestrichen werden, in denen das finite 'haben' im Abstand von höchstens 60 Zeichen (Output von Spraakbanken) und von beliebig vielen Wörtern vor 'inte' steht. Der reguläre Ausdruck für die vom ersten Programm noch nicht gestrichenen Textstellen lautet:

```
/(har |hade |Hade |Har )(\w|ä|ö|å)+ (\w|ä|ö|å)+ ((\w|ä|ö|å)+ )+inte varit/
```

Das heisst drei oder mehr Wörter vor 'inte' steht 'haben'. Aufgrund der Definition von Wort als "`(\w|ä|ö|å)+`" werden keine Satzzeichen überschritten. Es bleiben 38 Textstellen. Vier der fünf vom ersten Programm noch nicht gestrichenen Textstellen mit 'haben' in einem Hauptsatz werden nun eliminiert. In einem Fall wird ein Komma fälschlicherweise als Nebensatzanfang interpretiert. Kein einziger Nebensatz geht verloren. Das ist überraschend gut, in Anbetracht dessen, dass jede Art von Nebensätzen behandelt wird und innerhalb einer Umgebung von 60 Zeichen nur Satzzeichen als Teilsatzgrenzen erkannt werden können. Hätte man aber auch 'ha' als finite Form in den Filter aufgenommen, wäre der Nebensatz (4b) ohne Komma vor dem 'att' verlorengegangen. Diesen einen Fall hätte mein Programm zur att-Desambiguierung erkannt.

7.c) Fazit

Vermutlich liefert ein Programm, das mögliche Subjunktionen (deren Wortform) findet, die in einer Umgebung von 60 Zeichen vor einem Supinum stehen und nicht durch ein Satzzeichen von diesem getrennt sind, bereits befriedigende Ergebnisse. Man könnte anhand des Outputs einfach feststellen, ob zwischen der Subjunktion und dem Supinum ein finites Hilfsverb steht. Allerdings bleibt hier offen, ob die Supinumsformen möglichst vieler Verben zuverlässig erkannt werden können. Durch eine Kombination mit der Desambiguierung der möglichen Subjunktionen, wie sie das im Abschnitt 5 vorgestellte Programm vornimmt, könnte das Ergebnis verbessert werden. In diesem Sinn ist das in dieser Arbeit vorgestellte Programm auch brauchbar, wenn primär nach Verbformen in Nebensätzen gesucht wird.

8. Literatur

SAG: Svenska Akademiens grammatik. Ulf Teleman, Staffan Hellberg, Erik Andersson. Stockholm 1999.

Spraakbanken: Frei zugängliches Korpus der Universität Göteborg.[21] Darin unter 'Konkordanser' das Korpus 'äldre svenska romaner', das Romane aus dem 19. Jahrhundert enthält.

Rutishauser, Therese: Zur Frage der hochdeutschen Interferenzen in der schwedischen Syntax um 1700. Unveröffentlichte Lizenziatsarbeit. Universität Zürich 2001. (S.28-31).

[21]http://spraakbanken.gu.se genauer: http://spraakbanken.gu.se/lb/konk/

Anhang

A att-Desambiguierung

```
#!/usr/bin/perl                              #NSatt_,desamb
use Mac::StandardFile;
$file = StandardGetFile('','TEXT');
if ($file->sfGood()) {
     push(@ARGV, $file->sfFile());
} else {
     exit(1);
}
print @ARGV;
$filename = $ARGV[0];

open(INFILE, $filename);
open(OUTFILE, ">$filename.attDES,.out");
while (<INFILE>) {
     s/\n/ /;           #Entfernt Leerzeilen aufgrund von Doppelreturns
                        #am Abschnittende
     s/  / /g;
     s/\.\s/ PUNKT\.PUNKT /g;              #Leerschlag vor Satzzeichen
     s/\?\s/ FRAGE\?FRAGE /g;
     s/\!\s/ AUSRUF\!AUSRUF /g;
     s/(, |: |\; )/ $1/g;
     s/( Att | att )(han|hon|jag|det|den|de|någon) / NS_att $2 /
                        #Häufige Pronomina am Anfang der 'att'-Nebensätze
     s/, att /, NS_att /g;   #Komma vor 'att' Hinweis auf NS-Einleitung
     s/( på att )/ på INF_att /g;          #'Präfix'
     s/( Att | att )(((\w|ä|ö|å)+)a) / INF_att $2 /g;
     s/( Att | att )(se|bli|stå|få|fly|gå|dö|tro|förstå) / INF_att $2 /g;
                        #Infinitive direkt nach dem 'att' erkennen.
                        #Endung auf -a und Ausnahmeliste.
                        #Fehler, falls nach dem 'att' ein Wort steht,
                        #das auf -a endet aber kein Infinitiv ist.
                        #Bsp. 'hela Västerbotten' (='ganz Västerbotten')
     s/( Att | att )(inte )(((\w|ä|ö|å)+)a) / INF_att inte $3 /g;
     s/( Att | att )(inte )(se|bli|stå|få|fly|gå|dö|tro|förstå) / INF_att inte
$3 /g;
     s/( Att | att )/ NS_att /g;
                        #'att', die nicht bereits identifiziert wurden,
                        #werden hier als 'NS_att' klassifiziert.
     s/( Om | om )/ NS_om /g;
     s/( Som | som )/ NS_som /g;      #'om' und 'som' nicht differenziert
     s/( NS_|PUNKT |FRAGE |AUSRUF )/\n$1/g; #Zeilenumbruch an Teilsatzgrenzen.
     s/(: |\; |\, )/$1\n /g;
     print OUTFILE;
}
close(INFILE);
close(OUTFILE);
```

B. att-, om-, som- Nebensätze

```
#!/usr/bin/perl               #att,om,som_,desambEval
use Mac::StandardFile;
$file = StandardGetFile('','TEXT');
if ($file->sfGood()) {
     push(@ARGV, $file->sfFile());
} else {
     exit(1);
}
print @ARGV;
$filename = $ARGV[0];
```

```
open(INFILE, $filename);
open(OUTFILE, ">$filename.att.eval,.out");
while (<INFILE>) {
     s/\n/ /;
     s/  / /g;
     s/\.\s/ PUNKT\.PUNKT /g;     #Leerschlag vor Satzzeichen
     s/\?\s/ FRAGE\?FRAGE /g;
     s/\!\s/ AUSRUF\!AUSRUF /g;
     s/(, |: |\; )/ $1/g;
     s/( Att | att )(han|hon|jag|det|den|de|någon) / NS_att $2 /g;
                                  #Pronomina am Anfang der 'att'-Nebensätze
     s/( Att | att )(du |vi |ni )/ NS_att $2/g;
                          #Infinitive direkt nach dem 'att' erkennen:
                          #Ausnahmeliste:
     s/( Att | att )(se|bli|stå|få|fly|gå|dö|tro|förstå) / INF_att $2 /g;
     s/( Att | att )(inte )(se|bli|stå|få|fly|gå|dö|tro|förstå) / INF_att inte
$3 /g;
     s/( Att | att )(((\w|ä|ö|å)+) (PUNKT|FRAGE|AUSRUF))/ INFSATZLAENGE_att
$2/g;
                          #Ein Wort nach 'att' bereits Satzende
                          #bedeutet: Dieses Wort ist kein att-Nebensatz
                          #somit: ein Infinitiv.
                          #Einige Nichtinfinitive nicht nur defaultsmässig
                          #oder mit Hilfe des Kommas bestimmen:
     s/( Att | att )(en|ett) / NS_ARTatt $2 /g;
     s/( Att | att )(((\w|ä|ö|å)+)(n|t)) / NS_ARTatt $2 /g;
     s/(( s(ä|a)g| dröm| berätt)(a|e|)(r|t|tt|dde|de|)(|s)(| ,)) att /$1
NS_SAGatt /g;
     s/(( s(ä|a)g| berätt)(a|e|)(r|t|tt|dde|de|)(|s) (\w|ä|ö|å)+(| ,)) att /$1
NS_SAGatt /g;
     s/(( s(ä|a)g| berätt)(a|e|)(r|t|tt|dde|de|)(|s) (\w|ä|ö|å)+ (\w|ä|ö|å)+(|
,)) att /$1 NS_SAGatt /g;
     s/, att /, NS_KOMMAatt /g;
                          #Komma vor 'att' Hinweis auf NS-Einleitung
      s/( Att | att )(((\w|ä|ö|å)+ )+?(var|har|är|skulle)) / NS_FINatt $2 /g;
     s/( Att | att )(((\w|ä|ö|å)+ )+?((\w|ä|ö|å)(\w|ä|ö|å)+)(de|it)) /
NS_FINatt $2 /g;
           #(\w|ä|ö|å)(\w|ä|ö|å)+ mindestens zwei Buchstaben vor (de|it)
     s/( Att | att )(((\w|ä|ö|å)+)(a|as)) / INFEND_att $2 /g;
                          #Infinitive direkt nach dem 'att' erkennen:
                          #Endung auf -a. PROBLEM: wie Anhang A.
     s/( Att | att )(inte )(((\w|ä|ö|å)+)(a|as)) / INFEND_att inte $3 /g;
     s/( Att | att )(((\w|ä|ö|å)+)(e)) / NS_Eatt $2 /g;
                          #-e ist keine Infinitivendung
     s/( Att | att )/ NS_DEFAULTatt /g;
                          #'att', die nicht bereits identifiziert wurden,
                          #werden hier als 'NS_att' klassifiziert.
     s/, ((\w|ä|ö|å)+) (NS_(\w)*att) /, NS_$1$3 /g;
                          #", WORT NS_att " wird verknüpft
                          #Bsp. ", utan att ", ", så att ", ", därför att "
                          #=", ohne dass ", ", so dass ", ", deswegen dass "
                          #(=weil)

                          #Häufige Pronomina am Anfang der 'om'-Nebensätze
     s/( Om | om )(han |hon |jag |du |vi |ni |man )/ NS_NOMom $2/g;
     s/( Om | om )(honom |henne |mig |dig |sig |oss |er |dem )/ PREP_AKKom
$2/g;
     s/( Om | om )(((\w|ä|ö|å)+) (PUNKT|FRAGE|AUSRUF))/ PREP_SATZLAENGE1_om
$2/g;
                          #Ein Wort nach 'om' bereits Satzende
                          #bedeutet: Dieses Wort ist kein om-Nebensatz
                          #somit ev. temporales 'in' oder 'über'.
     s/( Om | om )(PUNKT|FRAGE|AUSRUF|\,|:|\;)/ PREP_SATZLAENGE0_om $2/g;
                          #Teilsatzende nach 'om'
                          #bedeutet: kein om-Nebensatz; somit ev. 'Präfix'.
     s/( Om | om )(NS_)/ PREP_SATZLAENGE?_om $2/g;
```

```
                        #Neuer Teilsatz nach 'om'
                        #bedeutet: kein om-Nebensatz; somit ev. 'Präfix'.
     s/ ((S|s)öder|(n|N)orr|öst|(v|V)äst|(v|V)änster|(h|H)öger) om / $1
PREP_POSITION_om /g;
     s/( Om | om )(en timme|en månad|dagen|en minut|en sekund|ett
år|natten|ett ögonblick|ett par ögonblick|en
vecka|vintern|våren|sommaren|hösten) / PREP_TEMP_om $2 /g;
     s/( Om | om )(((\w|ä|ö|å)+)
(timmar|månader|dagar|minuter|veckor|sekunder|år|nätter)) / PREP_TEMP_om $2
/g;                     #Temporale Ausdrücke nach 'om':
                        #'om' mit grosser Wahrscheinlichkeit Präposition.
     s/, om /, NS_KOMMAom /g;
                        #Komma vor 'om' Hinweis auf NS-Einleitung
     s/(( dröm| berätt| tigg| tal| be| tyck| bry)(a|e|)(r|t|tt|dde|de|)(|s))
om /$1 PREP_SPRECH_om /g;
     s/(( dröm| berätt| tigg| tal| be| tyck| bry)(a|e|)(r|t|tt|dde|de|)(|s)
(\w|ä|ö|å)+) om /$1 PREP_SPRECH_om /g;
     s/(( dröm| berätt| tigg| tal| be| tyck| bry)(a|e|)(r|t|tt|dde|de|)(|s)
(\w|ä|ö|å)+ (\w|ä|ö|å)+) om /$1 PREP_SPRECH_om /g;
     s/(( dröm| berätt| tigg| tal| be| tyck| bry)(a|e|)(r|t|tt|dde|de|)(|s)
(\w|ä|ö|å)+ (\w|ä|ö|å)+ (\w|ä|ö|å)+) om /$1 PREP_SPRECH_om /g;
     s/(( dröm| berätt| tigg| tal| be| tyck| bry)(a|e|)(r|t|tt|dde|de|)(|s)
(\w|ä|ö|å)+ (\w|ä|ö|å)+ (\w|ä|ö|å)+ (\w|ä|ö|å)+) om /$1 PREP_SPRECH_om /g;
     s/( Om | om )(((\w|ä|ö|å)+ )+?(var|har|är|skulle)) / NS_FINom $2 /g;
     s/( Om | om )(((\w|ä|ö|å)+ )+?((\w|ä|ö|å)+)(de|it)) / NS_FINom $2 /g;
     s/( Om | om )/ NS_DEFAULTom /g;
     s/(Om )/NS_DEFAULTom /g;    #Wegen Druckfehler im Text: Vor einem
                                 #'Om' ein unidentifizierbares Zeichen.
     s/, (och|som) (NS_(\w)*om) /, NS_$1$2 /g;
                                 #", (och|som) NS_om " wird verknüpft
                                 #=z.B.", und weil" oder ", wie wenn")

                                 #Finite Verbform nach 'som':
     s/( Som | som )(((\w|ä|ö|å|Å)+ )*?(var|har|är|skulle|stod)) / NS_FINsom
$2 /g;
     s/( Som | som )(((\w|ä|ö|å|Å)+ )*?((\w|ä|ö|å)+)(de|it|as|es)) / NS_FINsom
$2 /g;
     s/, som /, NS_KOMMAsom /g;  #Komma vor 'som' Hinweis auf NS-Einleitung
                                 #Kurze Vergleiche:
     s/(( samma | lika | så ))som /$1WIE0_som /g;
     s/(( samma | lika | så )(\w|ä|ö|å)+ )som /$1WIE1_som /g;
     s/(( samma | lika | så )(\w|ä|ö|å)+ (\w|ä|ö|å)+ )som /$1WIE2_som /g;
     s/(( samma | lika | så )(\w|ä|ö|å)+ (\w|ä|ö|å)+ (\w|ä|ö|å)+ )som
/$1WIE3_som /g;
     s/(( samma | lika | så )(\w|ä|ö|å)+ (\w|ä|ö|å)+ (\w|ä|ö|å)+ (\w|ä|ö|å)+
)som /$1WIE4_som /g;
     #s/( Som | som )(((\w|ä|ö|å)+ )*?(var|har|är|skulle)) / NS_FINsom $2 /g;
     #s/( Som | som )(((\w|ä|ö|å)+ )*?((\w|ä|ö|å)+)(de|it)) / NS_FINsom $2 /g;
     s/( Som | som )/ NS_DEFAULTsom /g;
     s/( NS_|PUNKT |FRAGE |AUSRUF )/\n$1/g; #Zeilenumbruch an Teilsatzgrenzen.
     s/(: |\; |, )/$1\n /g;
     print OUTFILE;
}
close(INFILE);
close(OUTFILE);
```

C. Experiment

```
$number = -2;
while (<INFILE>) {
  $number++;
                    #"inte varit" in $_
  if (/inte(,|\.|\?|\!) (varit |Varit)/) {
                    #Alternative: Satzzeichen zwischen 'inte' und 'varit'
                    #müssen gesucht werden. (Spraakbanken unterdrückt
                    #sie bei der Suche im Originaltext.)
  }
  else {
    if (/(har |hade |Hade |Har )inte varit/) {
                    #In der ersten Klammer können alle finiten Formen
                    #von 'haben' eingefügt werden.
                    #'varit' könnte durch ein anderes Supinum
                    #ersetzt werden.
      print OUTFILE0 "$number \t $_";
                    # "har inte varit" in $_
                    # ausgeschiedene Zeilen in .0out
    }
    else {
      if (/(har |hade |Hade |Har )(\w|ä|ö|å)+ inte varit/) {
        print OUTFILE1 "$number \t $_";
                    # "har WORT inte varit" in $_
                    # ausgeschiedene Zeilen in .1out
        }
        else {
          if (/(har |hade |Hade |Har )(\w|ä|ö|å)+ (\w|ä|ö|å)+ inte varit/) {
            print OUTFILE2 "$number \t $_";
                     # "har WORT WORT inte varit" in $_
                     # ausgeschiedene Zeilen in .2out
          }
          else {
            if (/(har |hade |Hade |Har )(\w|ä|ö|å)+ (\w|ä|ö|å)+ ((\w|ä|ö|å)+
)+inte varit/) {
              print OUTFILE3 "$number \t $_";
                     # "har WORT WORT WORT+ inte varit" in $_
                     # ausgeschiedene Zeilen in .3out
            }
            else {
              print OUTFILEres "$number \t $_";
                     # 'haben', falls vorhanden, weiter als
                     # 60 Zeichen von 'inte' entfernt,
                     # oder durch Satzzeichen von 'inte' getrennt.
            }
          }
        }
      }
    }
```