Maskininlärningsbaserad koreferensbestämning för nominalfraser applicerat på svenska texter

Magnus Danielsson

Examensarbete för 20 p, Institutionen för datavetenskap, Naturvetenskapliga fakulteten, Lunds universitet

Thesis for a diploma in computer science, 20 credit points, Department of Computer Science, Faculty of Science, Lund University

Maskininlärningsbaserad koreferensbestämning för nominalfraser applicerat på svenska texter

Sammanfattning

Denna examensrapport beskriver utvecklandet av en modul för maskininlärningsbaserad koreferensbestämning för nominalfraser. Modulen är en integrerad del i Carsim. Carsim är ett program för att omvandla texter på naturligt språk som beskriver en trafikolycka till en tredimensionell simulering av olyckan. Koreferensmodulen används i Carsim för att detektera de objekt som skall framträda i simuleringen. Den behandlar ofullständiga nominalfraser som inkluderar framförställda attribut och huvudord. Efterställda attribut saknas och endast en typ av inre nominalfraser finns definierad.

Modulen för koreferensbestämning är baserat på arbetet av Soon et al. (2001). I det arbetet används manuellt uppmärkta korpusar för att med hjälp av en beslutsträdsalgoritm automatiskt skapa en klassificerare. I tillägg till arbetet av Soon et al., som är ett domänoberoende system, har ett antal både domänberoende och domänoberoende egenskaper lagts till. Jag har gjort två, mig veterligen unika, utökningar av Soons algoritm. Dels används en kombination med ett filter med handkodade regler tillsammans med klassificeraren och dels används en konstruktion betecknad *egenskapsöverföring*. Egenskapsöverföring används för att kontinuerligt ändra semantiska egenskaper associerade till nominalfraserna i koreferenskedjorna till ett mer specifikt värde under klustringen. Utökningarna, relativt Soon et al., förbättrar resultaten markant.

Så vitt jag vet är detta det första helautomatiska systemet för koreferensbestämning avsett för svenska texter. Jag tror att koreferensmodulen relativt enkelt kan konverteras till norska eller danska på grund av stora likheter mellan de skandinaviska språken.

Machine learning based coreference resolution of noun phrases applied on Swedish texts

Abstract

This master’s thesis describes the development of a module for coreference resolution of noun phrases using a machine learning based approach. The module is an integrated part of Carsim. Carsim is a program that converts natural language texts describing a car accident into a 3D-simulation of the accident. The coreference module is used in Carsim to detect the objects to appear in the simulation. It considers partial noun phrases from the determiner to the headword. Post-modifiers are set aside and only one kind of inner noun phrases is defined.

The module for coreference resolution is based on the work by Soon et al (2001). It automatically induces a classifier with manually tagged corpora using a decision tree algorithm. In addition to the work by Soon et al., a domain independent system, a number of both domain dependent and domain independent features have been added. I have enhanced Soon's algorithm with two extensions, which are to my knowledge original. Firstly, a filter with handcoded rules is used together with the classifier. Secondly, a construction called *feature transfer* is implemented. Feature transfer is used to continuously change the values of semantic noun phrase features in the coreference chains during clustering. The extensions from Soon et al. improve the results dramatically.

As far as I know this is the first fully automatic system for coreference resolution for Swedish texts. I believe the coreference module should be easily portable to Norwegian or Danish because of great similarities between the Scandinavian languages.

Förord

Ett stort tack till min handledare Pierre Nugues för att ha introducerat mig till ett intressant ämne och för den tid, det engagemang och den konstruktiva kritik jag fått.

Jag vill också tacka Richard Johansson för all hjälp och det intresse han visat.

Magnus Danielsson

Lund, Januari 2005

Innehållsförteckning

Figurer

Tabeller

1 Inledning

1.1 Bakgrund

Carsim

Carsim (Johansson et al. 2004; Dupuy et al. 2001) är ett system som, utifrån en text som beskriver en trafikolycka, genererar en tredimensionell simulering av en olycksförloppet. Carsim består av två huvuddelar. Den första delen är ett system för informationsextrahering som utvinner språklig information från en text. Informationen sammanställs sedan i en formell beskrivning. Den andra delen använder beskrivningen för att skapa en grafisk representation av olyckan.

För att skapa en simulering identifieras tre typer av objekt från texten: scenobjekt, vägobjekt och händelser. Scenobjekt är statiska objekt i omgivningen som inte direkt är inblandade i olyckan, t.ex. väderförhållanden och vägtyper. Vägobjekt är de objekt som är inblandade i olyckan. Dessa kan t.ex. vara bilar, motorcyklar eller träd. Händelseobjekt omfattar de för simuleringen relevanta händelserna. Dessa kan t.ex. vara kollisioner eller omkörningar. Vidare ordnas händelserna i kronologisk ordning.

Vissa vägobjekt finns omnämnda på flera ställen i texten. För att få en korrekt simulering är det viktigt att avgöra vilka fraser i texten som representerar samma vägobjekt. Till detta används en modul för koreferensbestämning.

Koreferensbestämning

Koreferensbestämning för nominalfraser är en process för att avgöra vilka nominalfraser, som nämnts i en text, som refererar till samma entitet i världen (Ng 2002). Ett system för koreferensbestämning måste både identifiera nominalfraserna korrekt och extrahera tillräckligt med relevant information om nominalfraserna för att kunna avgöra koreferens. Varje system använder någon form av modell för att avgöra vilka nominalfraser som korefererar. Modellen kan baseras på logiska regler eller statistiska samband. Existerande system för maskinell koreferensbestämning använder antingen kunskapsbaserade eller korpusbaserade metoder. En kunskapsbaserad metod använder ett antal regler som skapats för hand för att avgöra koreferens. För en korpusbaserad metod används en mängd texter, en korpus, med korrekt koreferens uppmärkt som utgångspunkt. Utifrån information i dessa annoterade texter skapas en modell som används för koreferensbestämningen. *Maskininlärningsbaserad* koreferensbestämning är en korpusbaserad metod där modellen skapas automatiskt från texterna med någon maskininlärningsalgoritm.

Dagens system för koreferensbestämning har idag långt kvar till att uppnå resultat jämförbara med det en mänsklig annoterare kan prestera. Koreferensbestämning anses vara ett av de absolut svåraste problemen inom artificiell intelligens (Ng 2002). Svårigheten ligger framförallt i det faktum att det krävs detaljerad kunskap om världen i vissa fall:

Pälsen hängde över stolen. **Den** ser dyr ut, tänkte hon.

I exemplet är det trivialt för en läsare att avgöra att *pälsen* och *den* refererar till samma objekt. För dagens maskinella koreferenssystem är det dock omöjligt att med säkerhet avgöra att det inte är *stolen* som korefererar med *den*. I många andra fall kan dock koreferens med stor säkerhet avgöras utan detaljerad kunskap om världen. Starka indikatorer på koreferens kan t.ex. var grammatiska, lexikaliska eller semantiska likheter mellan nominalfraserna

1.2 Syfte

Syftet med examensarbetet är att skapa en modul i Carsim för koreferensbestämning av nominalfraser. Nominalfraserna är inte fullständiga. De inkluderar framförställda attribut och huvudord men efterställda attribut saknas. Dessutom finns endast en typ av inre nominalfraser definierad. Indata till modulen är svenska texter.

En mindre deluppgift består i att integrera resultatet i Carsim. Till detta används den utvecklade koreferensmodulen för att identifiera de nominalfraser som representerar objekt som skall framträda i simuleringen.

1.3 Metod

En utgångspunkt för koreferensmodulen är arbetet av Soon et al. (2001). Metoden är maskininlärningsbaserad, en beslutsträdsalgoritm används för att avgöra koreferens. I tillägg till metoderna i Soon et al. används även en filtermodul med handkodade regler samt en konstruktion betecknad egenskapsöverföring.

2 Koreferensbestämning för nominalfraser

I detta kapitel förklaras grundläggande begrepp som används vid bestämning av koreferens mellan nominalfraser. Även applikationer som kan dra nytta av koreferensbestämning samt tidigare arbeten beskrivs.

2.1 Nominalfraser

”Nominalfras är den sammanfattande termen för konstruktioner som fungerar på samma sätt som ett ensamt substantiv i syntaktiskt avseende.” (Hultman 2003, s. 204)

En nominalfras är uppbyggd kring ett huvudord, med eller utan bestämningar (Hultman 2003). Nominalfrasens bestämningar kallas *attribut* och kan vara framförställda eller efterställda. Bestämningarna ger ytterligare information om huvudordet. Hela nominalfrasen kallas också *maximal nominalfras* och huvudordet ensamt för *minimal nominalfras*.

Framförställda attribut	**Huvudord**	**Efterställda attribut**
–	mannen	i bilen
Den vita	bilen	som krockade med tåget
Den	vita	–
Djurgårdens	tränare	Niklas Wikegård
Mannens	bil	–
den stora gröna	husbilen	parkerad mitt i centrum
den andra kvinnans	Nissan Micra	–
–	den	–
Den	tredje	–

Tabell 2.1 Exempel på huvudord, framförställda och efterställda attribut för nominalfraser.

Varje nominalfras har ett huvudord som är lika med den minimala nominalfrasen. I typfallet är huvudordet ett substantiv: ”**bilen**”, ”den vita **bilen**”, ”**bilen** därborta”. Även andra typer av huvudord är möjliga, pronomen (”flera av **dem**”, ”**han**”), adjektiv (”den **vita**”), ordningstal (”den **tredje**”) eller ett egennamn (”**Drottning Silvia**”, ”en **Volvo 740**”, ”**Zlatan**”). Huvudordet kan bestå av fler än ett ord då det är ett egennamn. Även eventuell titel räknas här till namnet.

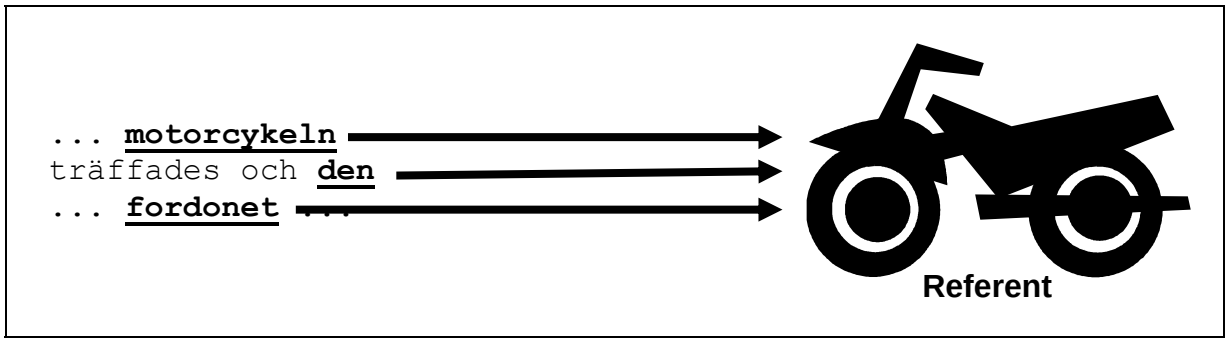
Framförställda attribut kan bestå av ett eller flera ord. Närmast framför huvudordet står *adjektivattribut*, adjektivfraser med eller utan bestämningar (”en **svart** katt”, ”**stora starka svenska** män”). Framför adjektivattributen står *genitivattribut* och *attribut som hör till pronomengruppen*. Ett genitivattribut är en nominalfras som står i genitiv (”**mannens** nya bil”). Ett attribut som tillhör pronomengruppen kan t.ex. vara: ”**den** bilen”, ”**den här/där** bilen” (demonstrativa), ”**vilken** bil?” (frågande) och ”**samtliga de** inblandade fordonen” (kvantitativa).

Efterställda attribut kan ha en mängd olika konstruktioner. Ingen närmare beskrivning av dessa ges då nominalfraserna som behandlas i det här arbetet nästan alltid saknar efterställda attribut.

I den fortsatta framställning benämns nominalfraser som är en del av en annan nominalfras som *inre nominalfraser*: I nominalfrasen ”den andra kvinnans Nissan Micra” är ”den andra kvinnans” en inre nominalfras.

2.2 Koreferensbestämning

I talat eller skrivet språk brukar inte samma nominalfras upprepas exakt för att referera till en entitet som introducerats tidigare i konversationen eller texten. Ofta ersätts den första representationen av entiteten med t.ex. ett pronomen eller kortform av nominalfrasen. Exempelvis kan samma entitet i ett dokument skrivas som ”Göran Persson”, ”han”, ”hans”, ”Göran”, ”Persson”, ”statsministern” eller ”Sveriges statsminister”. *Koreferensbestämning* för nominalfraser är en process för att avgöra om två eller flera nominalfraser hänför sig till samma referent, att bestämma alla nominalfraser som representerar ett visst objekt i världen (Ng 2002).



Figur 2.1 Tre korefererande nominalfraser har en gemensam referent.

Definition av koreferens

Koreferens mellan två nominalfraser, N1 och N2, definieras enligt: N1 och N2 korefererar ⇔ Referent(N1) = Referent(N2) (van Deemter & Kibble 2000). För två korefererande nominalfraser finns en *identitetsrelation* (MUC-7 1997). Relationen, betecknad IDENT, är en *ekvivalensrelation*. Det innebär att den är *symmetrisk* (N1 **IDENT** N2 ⇒ N2 **IDENT** N1), *transitiv* (N1 **IDENT** N2 & N2 **IDENT** N3 ⇒ N1 **IDENT** N3) och *reflexiv* (N **IDENT** N för alla N). Dessa egenskaper inducerar *ekvivalensklasser*. Varje element i varje relation ingår i exakt en ekvivalensklass och samtliga element i en ekvivalensklass korefererar. En ekvivalensklass, eller *koreferenskedja*, t.ex. N1 - N2 - N3 - N4, skapas ur ett antal korefererande par av nominalfraser, N1 - N2, N2 - N3, N3 - N4. Varje koreferenskedja består av samtliga nominalfraser som hänför sig till en gemensam referent.

Olyckan inträffade när **en bil** på väg nerför backen i riktning mot Jönköping gjorde en tvär omkörning. Föraren i **den framförvarande bilen**, **en Mazda**, tvingades göra en häftig undanmanöver och kom ut i mötande körfält. I **bilen** fanns tre personer. **Bilen** kolliderade med en Peugeot, med tre personer i. **Den omkörande bilen** försvann från olycksplatsen. Sent i går kväll hade polisen inga spår efter **den bilen** som smet.

Koreferenskedjor:
”en bil” – ”Den omkörande bilen” – ”den bilen”
”den framförvarande bilen” – ”en Mazda” – ”bilen” – ”Bilen”

Figur 2.2 Exempeltext (Hansson 2000) med koreferenskedjor. I det här exemplet visas de korefererande nominalfraserna som de extraheras i det här arbetet, utan efterställda attribut.

Bestämning av koreferens

Koreferensbestämning är ansett som ett svårt problem (Ng 2002). Det har funnits stor samstämmighet att svårigheterna beror på beroendet av sofistikerad semantisk kunskap samt kunskap om världen. Ingen informationskälla är ensam fullständigt pålitlig och en mängd olika indikatorer samverkar. Exempelvis är semantiskt kompatibla nominalfraser (t.ex. ”drottningen” och ”Silvia”) potentiellt korefererande men om de verkligen korefererar beror på kontexten. I figur 2.2 ges en illustration av detta. På två ställen i texten finns frasen ”tre personer” som trots att de är lika både till antal och semantisk klass (person) inte refererar till samma objekt. I andra fall, ”Nisse gick in på pizzerian och köpte en pizza. Den var god.”, behövs kunskap om världen. ”Den” korefererar med ”en pizza” och inte med ”pizzerian”. Vi vet att det var pizzan han tyckte var god, man äter sällan pizzerior. Vidare krävs olika strategier för olika typer av nominalfraser. Det är ofta svårare att bestämma koreferens för pronomen än för substantiv och egennamn.

Anafor och antecedent

Ordet *anafor* kommer av grekiskans *anaphora* (återföring, tillbakasyftning). Allmänt inom språkvetenskap betecknar anafor ett uttryck som refererar till ett uttryck som nämnts tidigare i en text. Det tidigare uttrycket betecknas *antecedent*.

Vid en *anaforisk relation* mellan två fraser beror anaforen på antecedenten för sin tolkning (för diskussion, se Ng 2002). Detta innebär att relationen mellan fraserna inte är symmetrisk. Den är inte heller transitiv eller reflexiv. Nominalfraserna (”Drottning Silvia”, ”hon”) har därmed en anaforisk relation medan (”hon”, Drottning Silvia”) inte har det. I båda exemplen är fraserna korefererande. I exemplet (”Drottning Silvia”, ”Sveriges drottning”) finns ingen anaforisk relation då ”Sveriges drottning” inte beror på ”Drottning Silvia” för sin tolkning. Fraserna korefererar dock. Målet vid *anaforabestämning* är att hitta en antecedent för varje nominalfras som beror på denna för sin tolkning. Vanligtvis är anaforen här ett pronomen. Anaforabestämning kan i princip ses som ett specialfall av koreferensbestämning: finns en anaforisk relation mellan två nominalfraser så korefererar de också. Det finns dock undantag, se exempel i nästa avsnitt, *Problem vid koreferensbestämning*. För en mer fullständig förklaring av skillnaderna mellan koreferens- och anaforabestämning se t.ex. Ng (2002) eller van Deemter & Kibble (2000).

Som visats finns det skillnader mellan anaforisk referens och koreferens. Av implementationsmässiga skäl används dock i princip samma metod vid maskinell koreferensbestämning som vid anaforabestämning. Det innebär att för varje ”anaforisk” nominalfras söker programmet efter en antecedent bland de framförvarande nominalfraserna, för att kontrollera om de korefererar. I den fortsatta framställningen används därför, i analogi med anaforabestämning, samma begrepp som vid anaforabestämning för alla nominalfraser i en koreferenskedja. För varje korefererande par kallas den nominalfras som förkommer först i texten för antecedent och den senare för anafor.

Problem vid koreferensbestämning

Att maskinellt bestämma koreferens är inte bara ett tekniskt problem. Det uppkommer i vissa fall mer filosofiska frågor om vad referens och koreferens innebär. Som tidigare nämnts så korefererar två nominalfraser om de har en gemensam referent. I MUC-7 (1997) finns rekommendationer, som använts i det här arbetet, för vilka nominalfraser som anses vara korefererande. Det finns fall där dessa rekommendationer inte följer definitionen av koreferens exakt. Följande diskussion är hämtad från van Deemter & Kibble (2000).

I frasen ”Varje gång jag såg **ett problem** så löste jag **det**” anses de två nominalfraserna vara korefererande i MUC-7. Detta trots att ”ett problem” inte kan sägas ha någon referent. Frasen avser varken någon enskild eller ändlig mängd entiteter. Däremot har nominalfraserna en anaforisk relation, ”det” beror på ”ett problem” för sin tolkning.

I frasen ”**Göran Persson**, tidigare **Sveriges finansminister**, blev nu vald till **Sveriges statsminister**” anses de tre nominalfraserna vara korefererande i MUC-7. Detta är dock inte rimligt enligt definitionen för koreferens. Då koreferens är en ekvivalensrelation skulle detta implicera att ”Sveriges finansminister” och ”Sveriges statsminister” var samma person.

I frasen "**Priset** sänktes från **2000kr** till **1500kr**" är det inte självklart vilka nominalfraser som korefererar. Naturligtvis kan inte ”2000kr” och ”1500kr” avse samma entitet. I MUC-7 anses ”Priset” och det senaste priset, ”1500kr”, vara korefererande. En annan möjlig lösning är att säga att ”Priset” i det här fallet är funktion från en tidpunkt till ett tal. Därmed skulle nominalfraserna inte anses vara korefererande.

Ytterligare problem uppstår i fall med potentiell koreferens: ”**Den ena bilen** kördes av den ensamme 20-åringen, och det var *troligen* **den bilen** som kom över på fel sida, uppger polisen”. Här avser ”Den ena bilen” och ”den bilen” samma entitet i vissa möjliga världar, men inte i andra (där polisen hade fel).

För vidare diskussion i ämnet se van Deemter & Kibble (2000). Det bör tilläggas att fall som de ovan beskrivna varit ovanliga i texterna som använts i det här arbetet.

Mening och referens – historisk bakgrund

Den tyske filosofen Gottlob Frege skriver i ”Über Sinn und Bedeutung” (”Om mening och referens”) (Frege 1892) om begreppet *referens*. Frege gör en distinktion mellan *mening* och *referens* för en fras. Fraserna ”morgonstjärnan” och ”aftonstjärnan” betecknar samma objekt, planeten Venus, och de korefererar därmed. Fraserna har samma referens men olika mening: ”den himlakropp som syns på morgonen” respektive ”den himlakropp som syns på kvällen”. Därmed är utsagan ”morgonstjärnan är samma sak som aftonstjärnan” informationsrik medan utsagan ”morgonstjärnan är samma sak som morgonstjärnan” inte ger någon information, den är en tautologi. Om olika fraser, med samma referens, har olika mening bidrar de var och en med information om referenten. Detta faktum utnyttjas i det här arbetet i en konstruktion kallad *egenskapsöverföring* (se 3.7).

2.3 Applikationer för koreferensbestämning

Här ges en sammanställning av olika typer av applikationer som kan dra nytta av koreferensbestämning. Om inget annat nämns är informationen hämtat från Ng (2002).

Informationsextrahering

Ett system för informationsextrahering tar en text från en given domän som indata och utvinner automatiskt information från texten. Koreferensbestämning används här för att sammanställa informationen som berör en viss entitet som finns omnämnd på olika platser i texten. Den här informationen är sedan till hjälp för att skapa en formell beskrivning där händelser kan associeras till relevanta objekt. Carsim använder ett system för informationsextrahering för att skapa en formell beskrivning av en trafikolycka omnämnd i en text.

Frågor och svar

Ett system för frågor och svar har till uppgift att besvara en fråga på naturligt språk med hjälp av en stor samling korpusar. En fråga som "Var föddes Mozart?" kan besvaras med "Han föddes i Salzberg" från en text som handlar om Mozart. I det fallet måste ett system för koreferensbestämning avgöra att "Mozart" och "Han" korefererar.

Textsummering

Summering av text hjälper användaren att få en bild av "viktig" information som finns i en stor textmassa. Summeringen är en sammanfattning av originaltexten utan redundant information. I exempelvis Microsoft Word finns numera en textsummeringsfunktion. Azzam et al. (1999) beskriver ett användningsområde för koreferensbestämning för textsummering. Deras utgångspunkt är det finns något centralt begrepp i en text som representeras av en koreferenskedja. Med hjälp av regler avgörs vilken koreferenskedja som är mest relevant och utifrån denna görs en sammanslagning av delmängder av de meningar som innehåller någon entitet från kedjan.

Koreferensbestämning över flera dokument

Målet är här att avgöra om två nominalfraser i olika dokument refererar till samma objekt. Koreferensbestämning över flera dokument är användbart i system för textsummering där informationen som skall sammanställas finns utspritt i flera dokument.

Maskinöversättning

System för maskinöversättning översätter en text på ett språk till ett annat språk. Mellan vissa språk är koreferensbestämning nödvändigt. Ett exempel är då ett pronomen, som korefererar med ett substantiv, behöver bestämmas för ett av språken. Här kan substantivets genus avgöra hur det anaforiska pronomenet och närliggande verb översätts.

2.4 Maskinell koreferensbestämning

Metoderna som används vid maskinell koreferensbestämning kan delas in i två huvudtyper: *kunskapsbaserade* och *korpusbaserade* (Ng 2002). Kunskapsbaserade metoder använder endast manuellt konstruerade regler för att avgöra koreferens. Korpusbaserade metoder utgår från en mängd texter, en korpus, med manuellt uppmärkta koreferenskedjor. Utifrån information i dessa annoterade texter skapas en modell som används för avgöra koreferens. Modellen kan skapas för hand, med statistiska metoder eller med maskininlärningsbaserade metoder.

Metoden för koreferensbestämning i det här arbetet är maskininlärningsbaserad med ett tillägg av ett fåtal enkla regler skapade för hand. En maskininlärningsbaserad metod utgår ifrån ett antal dokument med manuellt uppmärkta koreferenskedjor. Positiva och negativa träningsinstanser, antecedent-anaforpar, utvinns från de tränade texterna med någon metod. Varje par har en egenskapsvektor associerad till sig. Egenskapsvektorn innehåller ett antal attribut med t.ex. grammatisk eller semantisk information, som utvinns från paret. Utifrån egenskapsvektorerna för de positiva och negativa träningsexemplen induceras en klassificerare automatiskt med hjälp av en maskininlärningsalgoritm. Klassificeraren används sedan för att avgöra koreferens mellan två nominalfraser. En klustringsalgoritm skapar koreferenskedjor genom att länka samman par av korefererande nominalfraser där koreferensen bestämts av klassificeraren. Kedjorna bildar en partition på mängden korefererande nominalfraser.

Vad som skiljer olika arbeten för maskininlärningsbaserad koreferensbestämning är vilka kunskapskällor som används, vilken metod som används för att skapa träningsdata, vilken inlärningsbaserad algoritm som används samt vilken klustringsalgoritm som används.

2.5 En generisk algoritm för koreferensbestämning

Ng (2002) presenterar en genrisk algoritm som används för i princip all koreferens- och anaforabestämning. Algoritmens indata är en text utan restriktioner på implementationsspråket. De första tre stegen utförs på dokumentnivå medan de återstående stegen utförs på varje diskurselement i texten. Här presenteras de olika stegen med relevant beskrivning för koreferensbestämning för nominalfraser.

1 IDENTIFIERING AV DISKURSELEMENT	För algoritmer för koreferensbestämning innebär detta att alla nominalfraser i texten identifieras.
2 KARAKTÄRISERING AV DISKURSELEMENT	Först definieras en representation för nominalfrasen, dess karaktäristika. När en representation har bestämts är den andra uppgiften att beräkna informationen specificerad i representationen.
3 BESTÄMNING AV ANAFORER	Här bestäms om en nominalfras är anaforisk eller inte. Icke-anaforiska element saknar, per definition, en antecedent och algoritmen behöver då inte söka efter en. Vissa system använder inte detta steg och antar då att alla nominalfraser är potentiella anaforer.
4 GENERERING AV MÖJLIGA ANTECEDENTER	Då en nominalfras bestämts att vara anaforisk genereras här en lista med möjliga antecedenter. För de flesta algoritmer är detta alla nominalfraser före anaforen i dokumentet.
5 FILTRERING	Filtrering är en process för att ta bort vissa möjliga antecedenter genererade i 4 med hjälp av en mängd regler. Om någon av reglerna gäller tas elementet bort från listan med möjliga antecedenter.
6 POÄNGSÄTTNING eller RANKNING	Här används en algoritm för att poängsätta/rangordna varje möjlig antecedent efter hur sannolikt det är att kandidat-antecedenten - anaforen korefererar.
7 SÖKNING/KLUSTRING	I detta steg väljs en av de möjliga antecedenterna från kandidatlistan. Är listan tom väljs ingen antecedent. Har steg 6 utförts väljs det första/bästa elementet från listan. För fallet med koreferensbestämning innebär detta steg ofta att en enkellänkad klustringsalgoritm appliceras på varje anaforisk nominalfras. En partition skapas från de korefererande nominalfraserna i dokumentet med ett kluster (en koreferenskedja) för varje mängd nominalfraser som har en gemensam referent.

Figur 2.3 Generisk algoritm för koreferens och anaforabestämning.

Notera att steg 3, 5 och 6 inte behöver utföras, fast det är ovanligt att alla tre stegen saknas. Existerande algoritmer för koreferensbestämning skiljer sig åt hur de olika stegen implementeras. Vidare förklaring av de olika stegen i algoritmen, relevanta för det här arbetet, ges i kapitel 3.

2.6 Relaterade arbeten

Här presenteras tre olika arbeten som behandlar maskininlärningsbaserad koreferensbestämning. De visar att maskininlärningsbaserade system på senare år har uppnått fullt jämförbara resultat med kunskapsbaserade system. Arbetet av Soon et al. (2001) har fungerat som en utgångspunkt i det här arbetet.

McCarthy & Lehnert (1995) – Using Decision Trees for Coreference Resolution

McCarthy & Lehnert (1995) beskriver RESOLVE, ett system som är tänkt användas inom ett system för informationsextrahering. Systemet används för koreferensbestämning inom en specifik domän – texter om joint-venture (ett joint-venture är ett gemensamt projekt eller företag som drivs i samarbete mellan två eller flera företag). Metoden som används är maskininlärningsbaserad och en klassificerare skapas med beslutsträdsalgoritmen C4.5 (Quinlan 1993). Totalt använder RESOLVE åtta egenskaper i klassificerarens egenskapsvektor varav tre är domänspecifika för texter om joint-venture. T.ex. finns egenskaper som för både antecedent och anafor testar om de refererar till ett joint-venture. Av de domänoberoende egenskaperna är en positionell, befinner sig antecedent-anaforparet i samma mening? En är lexikalisk, delar de en gemensam nominalfras? Vidare finns egenskaper som avgör om antecedent eller anafor är ett egennamn. En egenskap, alias, definieras som sann om både antecedent och anafor är egennamn och ett av namnen är en substräng av det andra. Inga syntaktiska egenskaper har använts. Slutligen beskriver McCarthy & Lehnert en jämförelse mellan deras maskininlärningsbaserade metod och en kunskapsbaserad metod och rapporterar att den maskininlärningsbaserade metoden presterar bättre.

Soon et al. (2001) – A Machine Learning Approach to Coreference Resolution of Noun Phrases

Soon et al. (2001) beskriver ett domänoberoende system för koreferensbestämning. De använder beslutsträdsalgoritmen C5 (Rulequest Research 2004) för att skapa en klassificerare. Sammanlagt tolv olika egenskaper har använts för egenskapsvektorn. En av egenskaperna är lexikalisk, den jämför strängrepresentationen mellan antecedent och anafor. Åtta av egenskaperna är grammatiska: samma kön, samma numerus, apposition och olika nominalfrastyper. Vidare finns egenskaper för att avgöra om de tillhör samma semantiska klass, har samma alias (gemensamt namn) och en som avgör avståndet räknat i antalet meningar. Klassificeraren tränas med positiva och negativa träningsinstanser från texter med manuellt uppmärkta koreferenskedjor. Varje närmast intilliggande korefererande par används för att generera positiva träningsinstanser. För att minska ration mellan positiva och negativa träningsexempel, och därmed skevheten för klassdistributionen, används bara en delmängd av de negativa träningsexemplen. De skapas genom att för varje korefererande par generera negativa träningsexempel för alla mellanliggande nominalfraser i par med anaforen. En enkellänkad klustringsalgoritm som söker efter närmaste antecedent före anaforen används för att skapa koreferenskedjor. Resultatet indikerar att den lexikaliska egenskapen, egenskapen alias och egenskapen för apposition är starka indikationer på koreferens. Systemet är applicerat på två standardkorpusar för koreferensbestämning, MUC-6 (1995) och MUC-7 (1997). Resultatet är jämförbart med de bästa kunskapsbaserade systemen.

Ng & Cardie (2002a) - Improving Machine Learning Approaches to Coreference Resolution

Ng & Cardie (2002a) beskriver ett system som bygger på Soon et al. (2001). Systemet använder beslutsträdsalgoritmen C4.5 (Quinlan 1993). En utvidgning de gör från Soon et al. är att de utökar antalet egenskaper i egenskapsvektorn från 12 till 53. En ny positionell egenskap (samma stycke), åtta nya lexikaliska egenskaper (för strängjämförelser), fyra nya semantiska egenskaper och 26 nya grammatiska egenskaper har lagts till. Vidare har klustringsalgoritmen ändrats. Ng & Cardie använder en *bäst-först* klustringsalgoritm. Istället för att välja närmaste nominalfras som anses koreferera med anaforen används en annan metod. Den nominalfras, som av klassificeraren anses ha störst sannolikhet att koreferera av alla potentiellt koreferrerande nominalfraser, väljs som antecedent. Resultatet är att resultatet blir signifikant sämre, jämfört med Soon et al., då samtliga egenskaper i egenskapsvektorn används. Då istället 26 för hand utvalda egenskaper används förbättras både precision och F-värde signifikant.

3 Implementering av koreferensbestämning

En utgångspunkt för utvecklandet av modulen för koreferensbestämning har varit arbetet av Soon et al.[*] (2001). Systemet är maskininlärningsbaserat och bygger på en klassificerare i form av ett beslutsträd som skapats med beslutsträdsalgoritmen C5 (Rulequest Research 2004). Systemet använder tolv olika egenskaper i egenskapsvektorn.

Inte alla egenskaper från Soon et al. har implementerats och en mindre kraftfull beslutsträdsalgoritm, ID3 (Quinlan 1993), har använts här. I tillägg till metoderna presenterade i Soon et al. har en filtermodul med handkodade regler tillkommit. Egenskapen för semantisk klass i detta arbete är i högsta grad domänberoende, till skillnad från motsvarande egenskap i Soon et al. En annan viktig ändring är att egenskapen NUMBER (singular/plural) i Soon et al. ändrats till den betydligt mer kraftfulla egenskapen *antal objekt*. Vidare har en språkspecifik grammatisk egenskap för svenska, *grammatiskt genus* (utrum/neutrum), lagts till. Förutom dessa är ytterligare ett antal egenskaper implementerade. Klustringsalgoritmen är den samma, med tillägget att *egenskapsöverföring* under klustringen tillkommit. Fler jämförelser mellan detta arbete och arbetet av Soon et al. finns i kapitel 4.

I detta kapitel kommer metoderna som använts vid implementationen av modulen för koreferensbestämning beskrivas. Referenser till den generiska algoritmen i kapitel 2 kommer att ges för motsvarande steg här.

3.1 Indata

Som korpus för utveckling och testning har texter om trafikolyckor från olika svenska tidningar använts. Texterna går igenom en pipeline av språkmoduler implementerade i Carsim (Johansson et al. 2004; Dupuy et al. 2001). Texterna processas i tre olika språkmoduler innan resultatet presenteras för modulen för koreferensbestämning. Vad som utvinns av dessa tre moduler är *ordklasstaggar*, *namngivna entiteter* och *nominalfraser*.

[*] För mer information om arbetet av Soon et al., se under *Relaterade arbeten* i kapitel 2.

Ordklasstaggar

Carsim använder Granskas ordklasstaggare (Carlberger & Kann 1999). Taggaren annoterar, taggar, varje ord i en text med en märkningsetikett, en ordklasstagg. En ordklasstagg innehåller information om ordklass samt grammatisk information om ordet, t.ex. numerus och genus. Förutom ordklasstaggen märker Granska upp ordets *lemma*, ordet i grundform, samt vilken mening ordet tillhör.

Ord	Lemma	Ordklasstagg (ordklass + särdrag)
Skåpbilen	skåpbil	nn.utr.sin.def.nom
livshotande	livshotande	pc.prs.utr/neu.sin/plu.ind/def.nom
den	den	pn.utr.sin.def.sub/obj
sjukhuset	sjukhus	nn.neu.sin.def.nom
och	och	kn
timmar	timme	nn.utr.plu.ind.nom
Niklas Wikegård	niklas wikegård	sin.def.nom
tar	ta	vb.prs.akt
tom	tom	jj.pos.utr.sin.ind.nom

akt = aktiv form	nn = substantiv	pos = positiv
def = definit	nom = nominativ	prs = presens
ind = indefinit	obj = objektform	sin = singular
jj = adjektiv	pc = particip	sub = subjekt
kn = konjunktion	plu = plural	utr = utrum
neu = neutrum	pn = pronomen	vb = verb

Tabell 3.1 Några taggar uppmärkta av Granska med tillhörande förklaringar.

Varje ord tilldelas alla de tolkningar det kan ha då taggaren inte kan avgöra om ett ord står i t.ex. singular/plural eller indefinit/definit form. Ordklasstaggen inkluderar i de fallen ”sin/plu” respektive ”ind/def” för att indikera obestämbarhet.

Namngivna entiteter

Carsim märker upp vissa typer av egennamn med en etikett för vilken typ av objekt det representerar (Danielsson och Persson, 2004). Varje sådant textelement kallas för en *namngiven entitet*. Modulen för extrahering av namngivna entiteter är domänberoende för texter om trafikolyckor. Objekttyperna är anpassade för sådana texter. Följande etiketter finns definierade i modulen: BRAND (bilmärken), NAME (egennamn på personer), LOCATION (namngivna platser), AREA (landskap/länder), SQUARE (torg), ROAD (gator), STREET (vägar), HIGHWAY (motorvägar) och CITY (städer/stadsdelar).

Nominalfraser

Den viktigaste indatan kommer från modulen i Carsim som detekterar nominalfraser. Modulen taggar inte fullständiga nominalfraser. Något förenklat kan man säga att nominalfraser utan efterställda attribut men med framförställda attribut, då dessa inte är nominalfraser, taggas. Detta innebär att inga inre nominalfraser taggas.

3.2 Uppmärkningsbara element

Utifrån indatan definieras de textelement som är tänkbara att ingå i koreferenskedjor. På engelska används termen *markable* (Soon et al. 2001) för dessa element. I denna rapport kallas de fortsättningsvis för *uppmärkningsbara element*, eller bara *element*, om det tydligt framgår från sammanhanget vad som avses. Termen omfattar i det här arbetet dels nominalfrasen och dels en mängd egenskaper associerade till nominalfrasen.

Utgångspunkten för de uppmärkningsbara elementen har varit de nominalfraser som taggats av modulen för generering av nominalfraser i Carsim. Nominalfraserna från indatan saknar efterställda attribut och inre nominalfraser och det ligger inte inom ramen för detta arbete att utvidga definitionen av dessa i någon större omfattning. Ett par modifikationer har gjorts på nominalfraserna i indatan. En typ av inre nominalfraser har skapats och vissa nominalfraser har tagits bort, utifrån ordlistor.

Vad som tagits bort är t.ex. ensamma adjektiv utan determinerare som taggas som nominalfraser i stor utsträckning av Carsim. I andra fall taggas fraser felaktigt av ordklasstaggaren. T.ex. tolkas ”nordost” i ”nordost om Stockholm” felaktigt som ett substantiv och ordet blir därför uppmärkt som en nominalfras. I texterna, som i högsta grad är domänberoende, förekommer ett fåtal felaktiga fraser relativt högfrekvent i texterna. Därför är det lätt att med ordlistor filtrera ut dessa.

I MUC-7 Coreference Task Definition (MUC-7 1997) ges ett antal rekommendationer för vad som konstituerar ett uppmärkningsbart element. Rekommendationerna avser fullständiga nominalfraser med obegränsat antal inre nominalfraser för texter på engelska. Till viss del har dessa rekommendationer varit en utgångspunkt för definitionen av de uppmärkningsbara elementen i det här arbetet. En enkel utvidgning av nominalfraserna genererade av Carsim var att skapa en typ av inre nominalfraser. Detta gjordes för de fall då det framförställda attributet var i form av en nominalfras i genitiv.

Bortsett från språkskillnaden och skillnaden med inre nominalfraser är det några rekommendationer från MUC-7 som inte har följts i det här arbetet. Exempelvis finns inget stöd för att märka upp årtal eller procent (%). Vidare finns det antal typer av namngivna entiteter som inte märks upp vilket beror på begränsningar i modulen för generering av namngivna entiteter. För vissa typer av nominalfraser som saknas, eller är ovanliga, i de texter om bilolyckor som har använts för utvecklingen av det här arbetet har ingen kontroll gjorts att de märkts upp korrekt. Detta gäller exempelvis för frågepronomen.

Målsättningen har varit att en nominalfras i ett uppmärkningsbart element i det här arbetet skall definieras enligt följande regler:

- Allmänt gäller att alla nominalfraser t.o.m. huvudordet räknas som ett uppmärkningsbart element, med de restriktioner som beskrivs i följande regler.
- Huvudordet för det uppmärkningsbara elementet kan vara ett substantiv, adjektiv, ordningstal, namngiven entitet eller pronomen.
- Är huvudordet i en nominalfras ett personligt pronomen (han, jag, vi), demonstrativt pronomen (den, den här), reflexivt pronomen (sig, sin) eller possessivt pronomen (min, vår, er) räknas nominalfrasen som ett uppmärkningsbart element. Är nominalfrasens huvudord ett pronomen av någon annan typ räknas nominalfrasen inte som ett uppmärkningsbart element.
- Ensamma adjektiv eller ordningstal räknas inte som uppmärkningsbara element, vilket de däremot gör om de har ett framförställt attribut.
- För nominalfraser med efterställda attribut gäller att endast nominalfrasen t.o.m. huvudordet ingår. Undantaget är då det efterställda attributet endast består av en namngiven entitet. I det fallet räknas den namngivna entiteten inte som en inre nominalfras.
- För en nominalfras som inleds med ett framförställt attribut i form av en nominalfras i genitiv gäller: nominalfrasen i genitiv räknas som inre nominalfras om den saknar efterställt attribut. Inga andra typer av inre nominalfraser finns definierade.

[[den andra **kvinnans**] **Nissan Micra**]. *Nominalfras i genitiv blir inre nominalfras. Endast en typ av inre nominalfras används i det här arbetet, då huvudordet följer på en nominalfras i genitiv.*
[den vita **bilen**] som krockade med [**tåget**]. *Efterställt attribut tas ej med. Den fullständiga nominalfrasen skulle här omfatta hela texten med två inre nominalfraser.*
[[**Djurgårdens**] **tränare** Niklas Wikegård]. *Egennamn som efterställt attribut. Egennamn är den enda typen av efterställt attribut som används. Namnet blir dock inte en inre nominalfras.*
[Den **tredje**] klarade [**sig**] oskadd. *Ordningstal som huvudord.*
[**De**] var blåa och gula. *Ensamma adjektiv är ej uppmärkningsbara.*
[Den **blåa**] krockade med [den **gula**]. *Adjektiv med determinerare blir uppmärkta.*
Vem? [**Jag**]. *Frågepronomen är ej uppmärkningsbara.*

Figur 3.1 Uppmärkta nominalfraser enligt reglerna för uppmärkningsbara element i det här arbetet. Huvudord i fetstil.

3.3 Karaktärisering av uppmärkningsbara element - elementattribut

Detta steg motsvarar steg 2 i den generiska algoritmen i 2.4 (sid. 8). Här beräknas en mängd parametrar, *elementattribut*, för varje uppmärkningsbart element som senare behövs för att beräkna *egenskapsvektorer*. Den information som används för identifiering av elementattributen för ett element är nominalfrasens strängrepresentation, ingående ordklasstaggar och namngivna entiteter samt omgivande text. Varje element får ett värde efter vilken mening det förekommer i samt ett ordningsnummer. Ett inre element räknas som efterföljare till sitt föräldraelement.

Definition av elementattribut

Den första, och viktigaste, åtgärden är att identifiera huvudordet. Flertalet av de återstående elementattributen beräknas sedan utifrån huvudordet. Denna uppgift förenklas av att nominalfraserna saknar efterställda attribut (förutom namngivna entiteter). Det innebär att för element som saknar namngivna entiteter är huvudordet det sista ordet i frasen.

Tre olika lexikaliska attribut, strängrepresentationer, utvinns för elementet: Strängen utan determinerare (artiklar/demonstrativa pronomen), huvudordets strängrepresentation samt huvudordets strängrepresentation i grundform.

De flesta grammatiska egenskaper utvinns från huvudordets ordklasstagg. Intressanta egenskaper här är ordklass (substantiv, pronomen etc.), numerus (singular/plural), species (bestämd/obestämd form) och grammatiskt genus (utrum/neutrum). För dessa egenskaper gäller att de inte kan bestämmas entydigt av Granska i samtliga fall. Då blir egenskapsvärdena odefinierade. Elementet definieras som pronomen om huvudordet är ett pronomen och som en namngiven entitet om huvudordet är en namngiven entitet. Även etiketten för den namngivna entiteten blir här ett attribut, t.ex. NAME eller BRAND (se 3.1).

Elementets *ordningsnummer* (1, 2, 3, …) är definierat om ett ordningstal (förste/a, andre/a o.s.v.) förekommer innan huvudordet.

Elementets *namn* definieras med strängrepresentationen för en namngiven entitet: huvudordet om elementet är en namngiven entitet och med det efterställda attributet om detta finns och är en namngiven entitet. I övriga fall är namnet inte definierat.

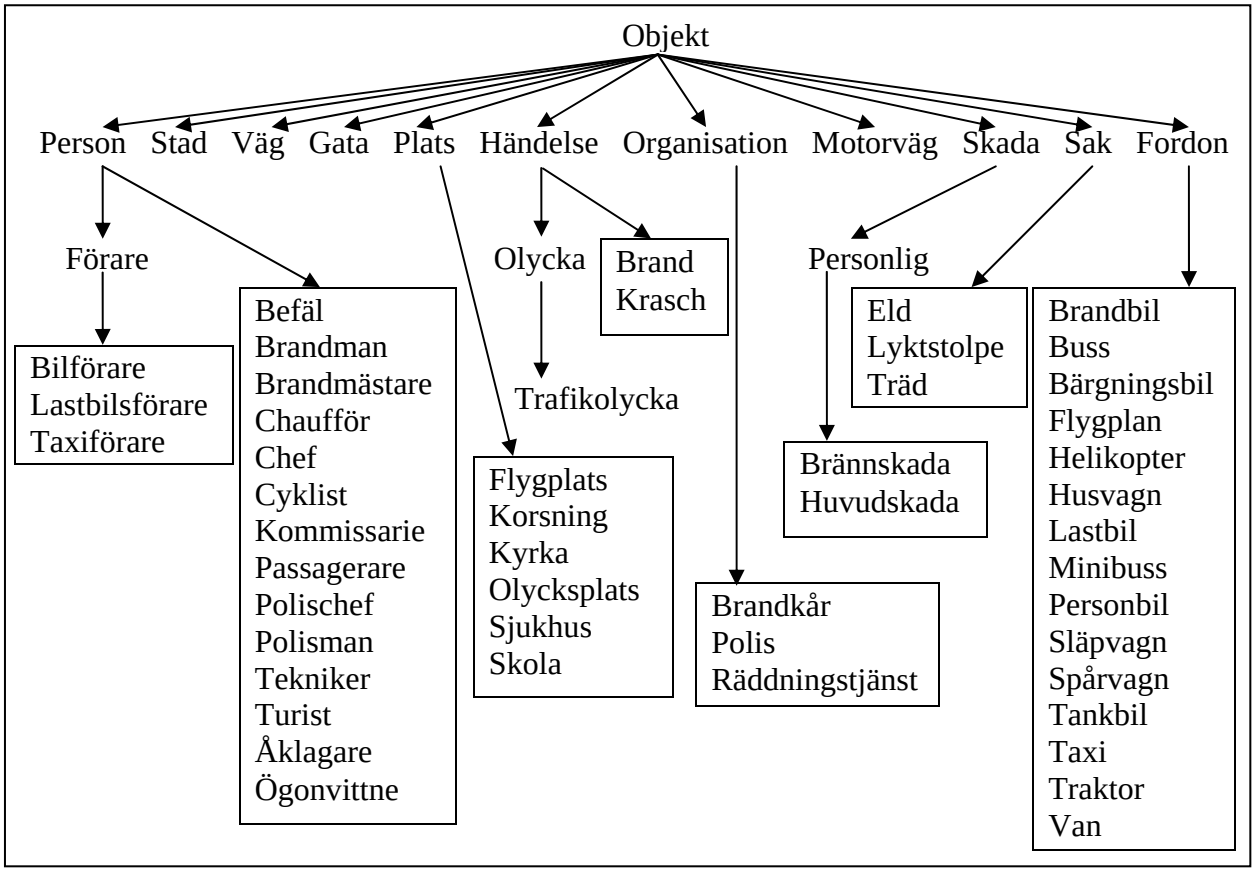
Antal objekt för elementet står för hur många objekt det representerar: ”en bil” (1), ”en av bilarna” (1), ”fyra bilar” (4), ”flera bilar” (>1), ”ingen av bilarna” (0), ”den tredje bilen” (1). Vidare kan värdet *okänt* ges om numerus är obestämt.

Egenskapen *nämnd tidigare* betecknar om det är sannolikt att objektet nämnts tidigare i texten. En nominalfras räknas som tidigare nämnt om det har ett ordningsnummer. I vissa fall, där nominalfrasen inleds med fraser som "en ...", "ett annat ..." eller "ytterligare en ..." räknas nominalfrasen som icke tidigare nämnd. I andra fall kan detta inte bestämmas.

Egenskapen *obestämd delmängd* sätts i falls som: ”en del av bilarna”, ”några av bilarna”, ”flertalet bilar”. Det avser delmängder av någon (troligtvis) tidigare nämnd mängd men där det inte kan avgöras vilken delmängd som avses.

Egenskapen *räknebar* sätts om huvudordet definieras som räknebar. Med ett räknebart substantiv menas att det inte kan delas (utan att förlora sina egenskaper) och kan förekomma i flera exemplar: ”en *bil*”, ”flera *bilar*”. Icke-räknebara substantiv betecknar sådant som kan delas (utan att förlora sina egenskaper) och inte kan förekomma i flera exemplar: ”de köpte *smör*”. Till icke-räknebara räknas även substantiv där inte något specifikt objekt avses: ”hon rökte *cigarr*”, ”han kör *bil*”. Endast nominalfraser bestående av ett substantiv i obestämd form kan definieras som icke-räknebar. Informationen i ordet före nominalfrasen används för att bestämma räknebarhet. Är ordet före t.ex. ett verb (han *kör* bil), adverb (han kör *inte* bil) eller en konjunktion (han kör bil *och* buss) räknas nominalfrasen som icke-räknebar.

Egenskapen *semantisk klass* tilldelar elementet en semantisk klass. Klasserna är uppbyggda i en hierarkisk trädstruktur (figur 3.2). De semantiska klasserna är i högsta grad domänberoende, de är anpassade för texter om trafikolyckor och fokus ligger på fordon och personer. Intentionen har varit att klasserna skall vara entydiga; att ett element tillhör en viss semantisk klass utesluter att elementet även tillhör en annan semantisk klass som inte är föregångare eller efterföljare till denna klass i klasshierarkin. Rotelement i trädet är *OBJEKT*. Det finns 11 barnelement till *OBJEKT* och totalt används 64 olika semantiska klasser i arbetet.



Figur 3.2 Den semantiska klasshierarkin.

För att identifiera semantisk klass används två metoder. Dels används en databas med reguljära uttryck och dels används eventuell namngiven entitet för elementet. Om elementet är en namngiven entitet sker en direkt översättning mellan etiketten för den namngivna entiteten och semantisk klass.

Tre elementattribut är specifika för en viss semantisk klass, *PERSON* och dess underklasser. Egenskaperna är *kön*, *ålderskategori* och *ålder* för personer. Ålderskategori kan ha värdena *barn* eller *vuxen* och ålder representeras av ett numeriskt värde för en persons ålder. Ålder kan utvinnas från fraser som ”en 20-årig man” eller ”tjugoåringen”. Ålderskategorin kan utvinnas på olika sätt. Först kontrolleras huvudordet från en ordlista. Exempelvis kategoriseras ”barnet”, ”ungdomarna” och ”flickan” som *barn* medan ”fadern”, ”pappan” och ”kvinnan” kategoriseras som *vuxen*. Finns ålder definierat för elementet räknas personer under 18 år som barn och personer som är 18 eller äldre som vuxna. En persons kön utvinns på liknande sätt utifrån ordlistor (”flickan”, ”pojken”, ”hon”, ”han”, o.s.v.). Finns *namnet* definierat för personen kontrolleras även ordlistor med vanliga svenska manliga och kvinnliga förnamn vilka är indikatorer på personens kön.

3.4 Egenskapsvektor

För inlärningsbaserad koreferensbestämning måste en mängd egenskaper definieras för att *klassificeraren* skall kunna bestämma om två uppmärkningsbara element korefererar eller inte. Dessa egenskaper bildar en *egenskapsvektor* som tilldelas varje par av uppmärkningsbara element som presenteras för klassificeraren.

Man kan kategorisera egenskaperna på olika sätt. En indelning har gjorts i lexikaliska, grammatiska, semantiska, positionella och övriga. Klassificeringen är här inte självklar för alla egenskaper. Vidare är vissa egenskaper *relationella* (jämförelser mellan en egenskap hos två element) eller *icke-relationella* (egenskap hos ett av de båda elementen). Totalt består egenskapsvektorn av 20 egenskaper. Varje värde för en egenskap baseras på det ena, eller båda, av två extraherade element där E1 är en potentiell antecedent och E2 anafor.

Typ	Egenskap	Relationella	Möjliga värden
Lexikaliska	Strängmatch	Ja	JA, NEJ
	Strängmatch för huvudord	Ja	JA, NEJ
	Strängmatch för huvudordets lemma	Ja	JA, NEJ
	Samma namn	Ja	JA, NEJ, OKÄNT
Grammatiska	E1 är pronomen	Nej	JA, NEJ
	E2 är pronomen	Nej	JA, NEJ
	E2 är i bestämd form	Nej	JA, NEJ, OKÄNT
	Samma numerus	Ja	JA, NEJ, OKÄNT
	Båda namngivna entiteter	Ja	JA, NEJ
	Samma grammatiska genus	Ja	JA, NEJ, OKÄNT
	Båda räknebara	Ja	JA, NEJ
Semantiska	Likhet för semantisk klass	Ja	JA, NEJ, OKÄNT
	Samma kön	Ja	JA, NEJ, OKÄNT
	Samma antal objekt	Ja	JA, NEJ, OKÄNT
Positionella	Avstånd	Ja	0, >0
	Är närmaste element	Ja	JA, NEJ
Övriga	E1 är ospecificerad delmängd	Nej	JA, NEJ
	E2 är ospecificerad delmängd	Nej	JA, NEJ
	E2 nämnd tidigare	Nej	JA, NEJ, OKÄNT
	Samma ordningsnummer	Ja	JA, NEJ, OKÄNT

Tabell 3.2 Egenskapsvektor.

Egenskaperna i egenskapsvektorn har definierats på följande sätt:

1 **Strängmatch:** Om strängrepresentationen för elementen, utan determinerare och omvandlat till gemener, är identisk ges värdet JA, annars NEJ.

2 **Strängmatch för huvudord:** Om strängrepresentationen för elementens huvudord, omvandlat till gemener, är identisk ges värdet JA, annars NEJ.

3 **Strängmatch för huvudordets lemma:** Om strängrepresentationen för elementens huvudords lemma är identisk ges värdet JA, annars NEJ.

4 **Samma namn:** Om båda elementen har elementattributet *namn* och dessa matchar enligt givna regler ges värdet JA. Matchar de inte ges värdet NEJ. Saknar ett eller båda elementen *namn* ges värdet OKÄNT. Reglerna för om namnen matchar är följande: Vid jämförelsen tas först eventuella avslutande ’s’ bort från namnen. Är typen för den namngivna entiteten hos båda elementen NAME (namn på personer) eller BRAND (namn på bilarmärken) anses namnen vara lika ifall ett av namnen är en substräng av det andra (t.ex. har ”Anders Perssons” och ”Persson” samma namn då ”Persson” är en substräng av ”Anders Perssons”). För övriga typer av namngivna entiteter anses namnen vara lika om namnen har identisk strängrepresentation..

5 **E1 är pronomen:** JA om E1 är ett pronomen, NEJ annars.

6 **E2 är pronomen:** JA om E2 är ett pronomen, NEJ annars.

7 **E2 är i bestämd form:** JA om E2 är i bestämd form, NEJ om det är i obestämd form, OKÄNT om elementets species inte kunde bestämmas.

8 **Samma numerus:** JA om elementen har samma numerus (singularis/pluralis), NEJ om de har olika numerus. Värdet OKÄNT ges om numerus för ett eller båda elementen inte kunde bestämmas

9 **Båda namngivna entiteter:** JA om båda är namngivna entiteter, NEJ annars.

10 **Samma grammatiska genus:** JA om elementen har samma grammatiska genus (utrum/neutrum), NEJ om de har olika grammatiska genus. Värdet OKÄNT ges om grammatiskt genus för ett eller båda elementen inte kunde bestämmas

11 **Båda räknebara:** JA om båda elementen kategoriseras som räknebara, NEJ annars.

12 **Likhet för semantisk klass:** JA om elementen tillhör samma semantiska klass, NEJ om de tillhör olika semantiska klasser. Kunde semantisk klass ej bestämmas för ett eller båda elementen ges värdet OKÄNT. Två element anses tillhöra samma semantiska klass om värdena för semantisk klass är identiska eller om den ena är förfader till den andra i den semantiska klasshierarkin.

13 **Samma kön:** JA om elementen har samma kön (man/kvinna), NEJ om de har olika kön. Värdet OKÄNT ges om kön för ett eller båda elementen inte kunde bestämmas. Värdet kan endast bestämmas om båda elementen tillhör den semantiska klassen *PERSON* eller dess underklasser.

14 **Samma antal objekt:** JA om elementen representerar samma antal objekt, NEJ om de representerar olika antal objekt. Värdet OKÄNT ges då detta inte kunde bestämmas. Följande regler har applicerats (i nämnda ordning): Är värdet för antal objekt 0 för ett eller båda elementen returneras NEJ (en inkorrekt definition som dock, oftast korrekt, förhindrar koreferens). Är värdet för antal objekt okänt för ett eller båda elementen returneras OKÄNT. Är antal objekt >1 för ett element returneras OKÄNT om det andra elementets antal objekt är >1 och NEJ för övriga värden på det andra elementet. Är antal objekt definierat med ett specificerat numeriskt värde för båda elementen returneras JA om de är lika, NEJ annars.

15 **Avstånd:** Värdet 0 ges om elementen finns i samma mening, >0 annars.

16 **Är närmaste element:** Om E2 har ordningsnumret *n* returneras JA om E1 har ordningsnumret *n-1*, NEJ annars.

17 **E1 är ospecificerad delmängd:** JA om E1 är definierad som en ospecificerad delmängd, NEJ annars.

18 **E2 är ospecificerad delmängd:** JA om E2 är definierad som en ospecificerad delmängd, NEJ annars.

19 **E2 nämnd tidigare:** JA om E2 är definierad som tidigare nämnd, NEJ annars.

20 **Samma ordningsnummer:** JA om elementen har samma ordningsnummer, NEJ om de har olika ordningsnummer. Värdet OKÄNT ges om ordningsnummer för ett eller båda elementen inte kunde bestämmas.

3.5 Filtrering

Detta steg motsvarar steg 5 i den generiska algoritmen i 2.4 (sid. 8). Filtrering innebär att mängd handkodade regler används för att för förhindra koreferens mellan två uppmärkningsbara element. Primärt är syftet med ett sådant filter att öka precision för andelen korrekt uppmärkta element. Reglerna är enkla, de utnyttjar bara värden från egenskapsvektorn och elementattributen. Intuitivt verkar de semantiska egenskaperna vara lämpliga kandidater. Exempelvis kan en man och en kvinna inte referera till samma objekt. Dock har även andra regler som inte med säkerhet kan sägas utesluta koreferens använts. I de fallen får man väga nackdelarna med att vissa korefererande antecedent-anaforpar hindras av filtret mot att det totala resultatet förbättras. T.ex. kommer filtervillkoret *Samma grammatiska genus = NEJ* förhindra att ”huset” (neutrum) och ”byggnaden” (utrum) kan koreferera.

Totalt består filtret av tolv regler varav tio använder egenskaper ur egenskapsvektorn. De övriga två, *samma ålderskategori* och *samma ålder*, får sina värden genom jämförelser av elementens motsvarande elementattribut. Vid följande villkor förhindras koreferens för ett antecedent-anaforpar:

1 Samma antal objekt = NEJ
2 Samma grammatiska genus = NEJ
3 Likhet för semantisk klass = NEJ
4 Samma kön = NEJ
5 Samma namn = NEJ
6 Samma ordningsnummer = NEJ
7 E1 är ospecificerad delmängd = JA
8 E2 är ospecificerad delmängd = JA
9 Båda räknebara = NEJ
10 E2 nämnd tidigare = NEJ
11 Samma ålderskategori = NEJ
12 Samma ålder = NEJ

3.6 Klassificerare

För varje antecedent-anaforpar behövs en klassificerare för att avgöra koreferens. Exempelvis kan varje par klassificeras med ett numeriskt värde vilket representerar sannolikheten för koreferens[*]. Denna metod är lämplig då man använder en *bäst-först* klustringsalgoritm (Ng & Cardie 2002b). I det här arbetet klassificeras varje par i en av två möjliga klasser; varje antecedent-anaforpar klassificeras som antingen *korefererande* eller *icke-korefererande*.

Här är klassificerarens representationsform ett beslutsträd. Ett beslutsträd är en trädstruktur där varje intern nod svarar mot ett attribut vars möjliga värden vart och ett leder till en ny nod och lövnodernas värde representerar en klass. Ett exempel klassificeras genom att, utgående från rotnoden, upprepat gå vidare till noden vars attributvärde motsvarar exemplets tills en lövnod har nåtts. Exemplets klass blir här den som anges av lövnoden.

```
if (Strängmatch == Ja) {
          ...
}
if (Strängmatch == Nej) {
          if (Avstånd == 0) {
                    if (E2 är pronomen == Ja)
                              return JA;  ← Koreferens
                    if (E2 är pronomen == Nej) {
                              return NEJ; ← Ingen koreferens
                    }
          if (Avstånd == >1) {
                    ...
          }
          ...
}
```

Figur 3.3 Delmängd av ett beslutsträd i en javalik struktur. Varje intern nod svarar mot ett attribut i egenskapsvektorn och varje löv svarar JA eller NEJ på frågan om koreferens för det antecedent-anaforpar som testas.

Generering av beslutsträd

Beslutsträdet induceras utifrån en mängd förklassificerade exempel med hjälp av en beslutsträdsalgoritm, ID3 (Quinlan 1993). Beslutsträdsalgoritmen tar två mängder med exempeldata innehållande par av element från de båda klasserna, *positiva* (korefererande) och *negativa* (icke-korefererande), som indata. Datan består av parets egenskapsvektor.

[*] I Ng & Cardie (2002a) används denna metod tillsammans med en bäst-först klustringsalgoritm.

För att generera indatan till beslutsträdsalgoritmen, positiva och negativa träningsexempel, behövs en mängd texter med koreferenskedjorna manuellt uppmärkta (Soon et al. 2001). För en manuellt uppmärkt koreferenskedja, t.ex. E1 – E2 – E3 – E4, används endast par av element som är närmaste grannar för att generera positiva träningsexempel (E1 – E2, E2 – E3, E3 – E4). På detta sätt genereras positiva träningsexempel genom att extrahera alla sådana par från alla texter med manuellt uppmärkta koreferenskedjor. Negativa träningsexempel genereras på följande sätt. Mellan varje antecedent-anaforpar finns ibland element som inte ingår någon koreferenskedja eller ingår i en annan koreferenskedja. Om elementen x, y och A2 är placerade mellan E1 och E2 genereras de negativa träningsexemplen x – E2, y – E2 och A2 – E2 d.v.s. alla element som ligger mellan en antecedent och en anafor bildar tillsammans med anaforen negativa träningsexempel. I exemplet ingår x och y inte i någon koreferenskedja medan A2 ingår i en annan koreferenskedja.

Endast ett litet antal av alla potentiella träningsexempel används för att skapa klassificeraren. Anledningen att inte ta med alla negativa träningsexempel är att skevhet i klassdistributionen bör minskas. Cardie & Howe (1997) beskriver problemet att om en av klasserna (i det här fallet ingen koreferens) som klassificeraren tränas på är i stor majoritet försämras resultatet signifikant för minoritetsklassen (koreferens). Det är uppenbart att de allra flesta träningsexempel är negativa, koreferens är en ovanlig relation. Exempelvis är endast ca. 2% av instanserna i MUC-6 och MUC-7 positiva (Ng 2002). Urvalsmetoden för negativa träningsinstanser minskar skevheten kraftigt. Urvalsmetoden för positiva träningsexempel är anpassad efter klustringsalgoritmen som används (se 3.7).

ID3

ID3 (Quinlan 1993) är en beslutsträdsalgoritm som bygger på entropibegreppet (Shannon 1948). Utifrån ett begränsat antal träningsinstanser skapar algoritmen ett beslutsträd som klassificerar varje framtida exempel som presenteras för det. Vid skapandet av trädet arbetar algoritmen rekursivt, för varje subträd delar den de återstående exemplen så att det återstår färre träningsinstanser och ett attribut mindre. Delningskriteriet som används kallas *information gain*, vilket strävar efter att minska entropin maximalt i trädet. Varje lövnod får ett värde mellan 0 och 1 vilket motsvarar sannolikheten för koreferens i det här arbetet. Detta värde översätts till koreferens vid värden >0.5 och icke-koreferens annars.

ID3 har vissa tillkortakommanden jämfört med sin efterföljare C4.5 (Quinlan 1993). För det här arbetet är det primärt två utvidgningar av ID3 som skulle kunna förbättra resultatet: *gain ratio* och *beskärning* (eng. pruning). *Gain ratio* är ett mått som ersätter delningskriteriet *information gain* i ID3 och innebär bättre klassificering för attribut med många värden. *Beskärning* används för att ta bort grenar på beslutsträdet som försämrar resultatet.

3.7 Klustringsalgoritm – generering av koreferenskedjor

Detta motsvarar steg 7 i den generiska algoritmen i 2.4 (sid. 8). I tidigare steg har och uppmärkningsbara element identifierats och karaktäriserats. Vidare har en klassificerare och ett filter med regler skapats. Nästa steg är att med hjälp av klassificeraren skapa koreferenskedjor av de element som representerar samma referent.

Syftet med en klustringsalgoritm är att partitionera en mängd objekt i kluster med hjälp av något mått på likhet (Pantel 2003). För fallet med maskininlärningsbaserad koreferensbestämning har det i senare arbeten varit vanligt att använda en *enkellänkad klustringsalgoritm* (Ng & Cardie 2002b). I t.ex. Soon et al. (2001), liksom i det här arbetet har en *enkellänkad klustringsalgoritm* med *höger-till-vänster-sökning* (Ng 2002) använts.

Klustringsalgoritmen använder klassificeraren samt en sökalgoritm för att skapa koreferenskedjor. Varje element i texten, utom det första (som inte kan vara en anafor), testas med närmast framförvarande element för koreferens av klassificeraren (om det inte stoppats av filtret). Undantaget är inre element; är det ena elementet inre element till det andra kan de inte koreferera. Korefererar elementen och inget av elementen sedan tidigare finns i en koreferenskedja skapas en kedja av de båda elementen. Om de korefererar och ett av elementen finns i en koreferenskedja sedan tidigare läggs det andra elementet till kedjan. Korefererar elementen inte så fortsätter sökandet bakåt i texten tills en antecedent hittats eller inget av de framförvarande elementen korefererar med aktuellt element. Sedan testas nästa element på samma sätt tills sista elementet i texten har undersökts. Slutresultatet blir en partition på de korefererande elementen där elementen i varje kluster har en gemensam referent.

((Mannens)**E2** bil) **E1** krockade med (ett träd) **E3**. (Mannen) **E4** klarade (sig) **E5** med (lindriga skador) **E6** medan (bilen) **E7** fick (svåra plåtskador) **E8**.

Antecedent	**Anafor**	**Koreferens?**
(Mannens)**E2**	(ett träd) **E3**	Nej
(Mannens bil) **E1**	(ett träd) **E3**	Nej
(ett träd) **E3**	(Mannen) **E4**	Nej
(Mannens)**E2**	(Mannen) **E4**	Ja
(Mannen) **E4**	(sig) **E5**	Ja
…	...	…
(Mannens bil) **E1**	(svåra plåtskador) **E8**	Nej

Kedja 1: (Mannens)**E2** - (Mannen) **E4** - (sig) **E5**
Kedja 2: (Mannens bil) **E1** - (bilen) **E7**

Tabell 3.3 Exempel på ordningen elementen testas i av klustringsalgoritmen. Först testas E3 - E2 för koreferens. E2 - E1 testas inte då E2 är ett inre element till E1.

Egenskapsöverföring

Alla element i en koreferenskedja representerar samma objekt, deras referent. Detta innebär att alla kedjans element bör ha vissa egenskaper gemensamt, de egenskaper som referenten har. Är referenten t.ex. en grupp på tre män, är egenskaper för referenten att den består av tre objekt, objekten är personer och de har alla könet man. Ett elements semantiska egenskaper och eventuellt namn kan sägas vara egenskaper hos referenten medan t.ex. grammatiska egenskaper inte är det.

En idé är att vid varje tillfälle ett antecedent-anaforpar identifierats låta de egenskaper som kan sägas tillhöra referenten fortplanta sig till samtliga element i kedjan. Vid varje sådant tillfälle ersätts varje egenskapsvärde, hos alla ingående element i kedjan, med det *mest specifika* värdet för den egenskapen som existerar hos något element i kedjan. Några exempel: Är semantisk klass OBJEKT/FORDON för ett element och OBJEKT/FORDON/BIL för ett annat, kommer elementen att få semantisk klass OBJEKT/FORDON/BIL. Är antal objekt ”>1” för ett element och ”3” för ett annat kommer elementen att få antal objekt ”3”. Är namnet för ett element ”Kalle” och ”Kalle Andersson” för ett annat kommer elementen att få det längsta namnet, ”Kalle Andersson”. På detta sätt kan varje element bli mer specificerat för varje element som läggs till kedjan. Vid nästa tillfälle ett element testas för koreferens med ett element i kedjan ökar sannolikheten att koreferens kan bestämmas korrekt.

Jag har valt att kalla konstruktionen för *egenskapsöverföring*. Grundtanken har varit att ju mer specifik kunskap det finns om ett element ju lättare blir det för klassificeraren att bestämma koreferens korrekt. Det visar sig att egenskapsöverföringen har signifikant effekt på *precisionen* (se kapitel 4). Vid ett test ökar precisionsvärdet från 74.4%, utan egenskapsöverföring, till 79.5% då egenskapsöverföring används.

Det mest specifika värdet bland elementen i kedjan överförs för nedanstående sju elementattribut. Som synes är ett av attributen grammatiskt, *grammatiskt genus*. Därmed kan det inte sägas att attributet är en egenskap hos referenten. Preliminära tester har dock gett indikationer på attributet förbättrar resultatet och får därför finnas kvar:

- Semantisk klass
- Antal objekt
- Namn
- Kön
- Grammatiskt genus
- Ålder
- Ålderskategori

3.8 Portabilitet

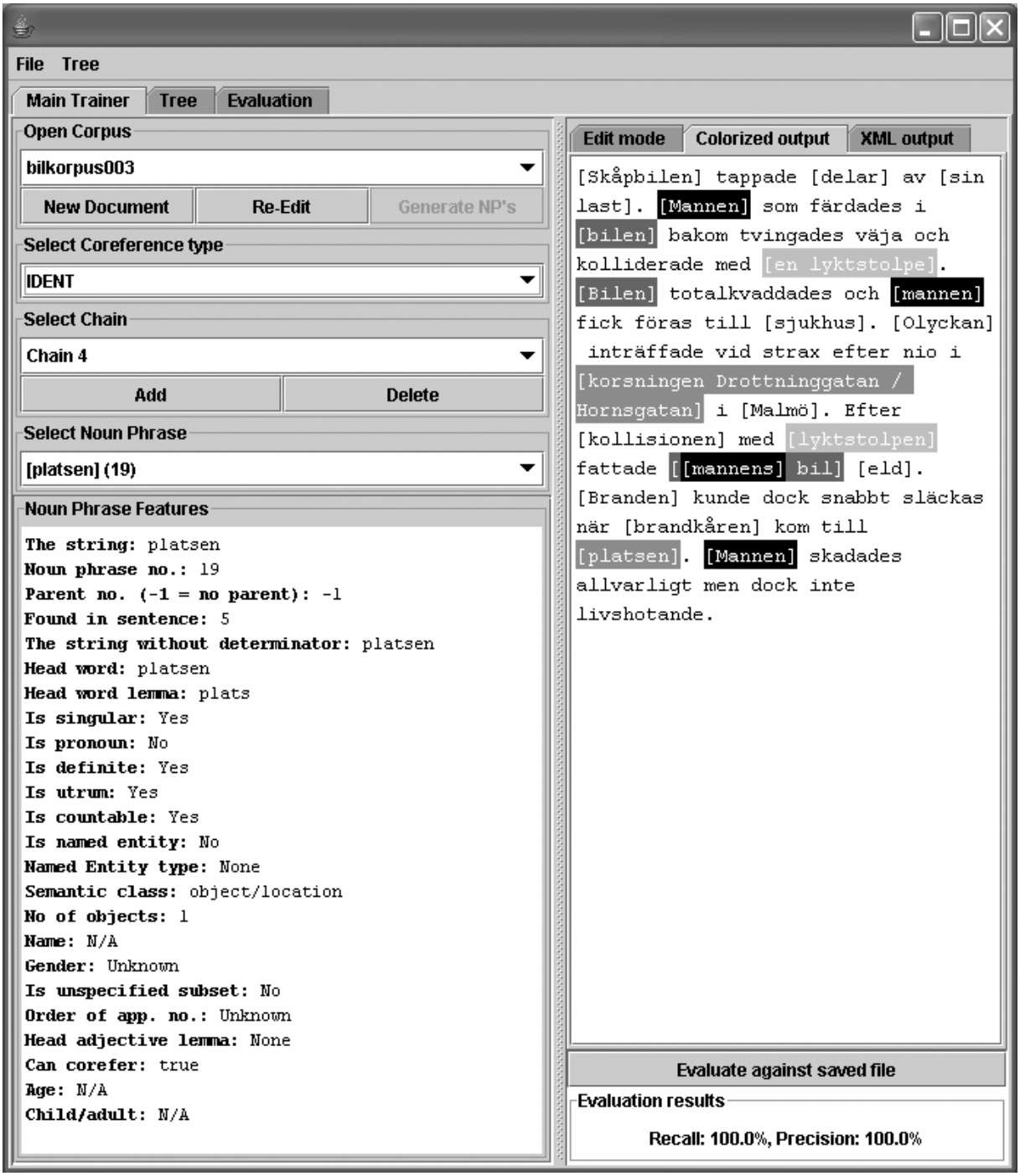
Jag tror att många aspekter i systemet är universella, att de kan användas för andra domäner och för andra språk. Framförallt egenskapsöverföringen torde kunna användas för andra språk med små, eller inga, modifikationer. Den språkspecifika kärnan i koreferensmodulen är elementattributen, och i viss mån egenskapsvektorn. Lättast bör det vara att konvertera koreferensmodulen till de övriga skandinaviska språken, danska och norska, som har mycket stora likheter med svenskan. Att språken har likartad ordföljd underlättar. Med enkla medel kan de reguljära uttrycken som används för att bestämma semantisk klass och personnamnen som används för att bestämma kön översättas. Elementattributen *nämnd tidigare* och *obestämd delmängd* bör fungera på andra skandinaviska språk genom att översätta ett fåtal nyckelord som använts för att bestämma dessa. För de skandinaviska språken bör egenskapsvektorn och egenskapsöverföringen kunna användas utan modifikationer. För ett fullt fungerande system för koreferensbestämning för andra språk krävs dock även språkspecifika moduler för identifiering av nominalfraser och namngivna entiteter samt en ordklasstaggare för bestämning av de grammatiska egenskaperna.

3.9 Utvecklingsmetodologi

En stor del av arbetet har bestått i att utveckla en applikation till hjälp för utvecklandet av koreferensmodulen. Applikationen har ett grafiskt gränssnitt och målsättningen har varit att man på ett snabbt och överskådligt sätt skall kunna testa idéer och se resultatet av dessa. På samma sätt som för koreferensmodulen används moduler i Carsim för att generera ordklasstaggar, namngivna entiteter och nominalfraser. Följande funktioner finns implementerade:

- Möjlighet att skapa uppmärkningsbara element utifrån en text på svenska. Texterna kan laddas från fil eller skrivas in i ett textfönster.
- Möjlighet att skapa träningsdokument med uppmärkta koreferenskedjor genom att klicka på nominalfraser med muspekaren.
- Möjlighet att kunna se vilka elementattribut som programmet extraherat för varje nominalfras genom att klicka på nominalfrasen med muspekaren.
- Skapande av beslutsträd från de tränade dokumenten. Vilka dokument och vilka parametrar i egenskapsvektorn som skall ingå är valbart. Man kan även få beslutsträdet utskrivet.
- Möjlighet att applicera beslutsträdet på en, eller flera, valfria texter. Här går det att välja vilka parametrar i filtret och egenskapsöverföringen som skall användas. Det går också att se resultatet i form olikfärgade nominalfraser med en unik färg för varje koreferenskedja.
- Utvärdering av resultatet av effekten av ett visst beslutsträd på valfri delmängd av de dokument som har manuellt uppmärkta koreferenskedjor. Precision och täckning räknas ut.

Applikationen har varit en ovärderlig hjälp vid utvecklande av koreferensmodulen. Framförallt att man direkt kunnat se grafiskt hur en viss parameterkonfiguration påverkat resultatet har varit till nytta. Det har skyndat på utvecklingen av nya egenskaper och förändring av definitionen för andra egenskaper.



Figur 3.4 Applikation för koreferensbestämning. Ett beslutsträd har applicerats på en text och koreferenskedjorna syns som mängder av olikfärgade nominalfraser. Utvärderingsresultat jämfört med ett sparat, manuellt uppmärkt, dokument visas. Elementattribut för den senast klickade nominalfrasen visas. Texten kommer ursprungligen från Tagesson (2002).

4 Utvärdering av koreferensbestämning

I detta kapitel kommer utvärderingar att göras för koreferensbestämningen på ett antal texter om trafikolyckor. Metoden för utvärderingen beskrivs. Vidare utförs ett antal tester med olika parameterinställningar och olika antal träningsdokument.

4.1 Metod för utvärdering av koreferensbestämning

Annotering

I MUC-7 Coreference Task Definition (MUC-7 1997) finns ett schema för annotering av koreferenskedjor i SGML. En delmängd av detta schema används i detta arbete för representationen av koreferenskedjor vid utvärderingen (se figur 4.1). Tre attribut används, ID, REF och TYPE. Varje uppmärkt uttryckt har ett unikt ID-värde och REF-värdet betecknar att uttrycket är anafor till uttrycket med motsvarande ID-värde. TYPE kan endast ha värdet IDENT vilket står för identitetsrelationen presenterad i 2.2. För utvärderingen representeras en texts koreferenskedjor i form av ett XML-dokument. Även texten, namngivna entiteter och nominalfraser representeras av XML-dokument, totalt fyra dokument för varje text har använts vid utvärderingen.

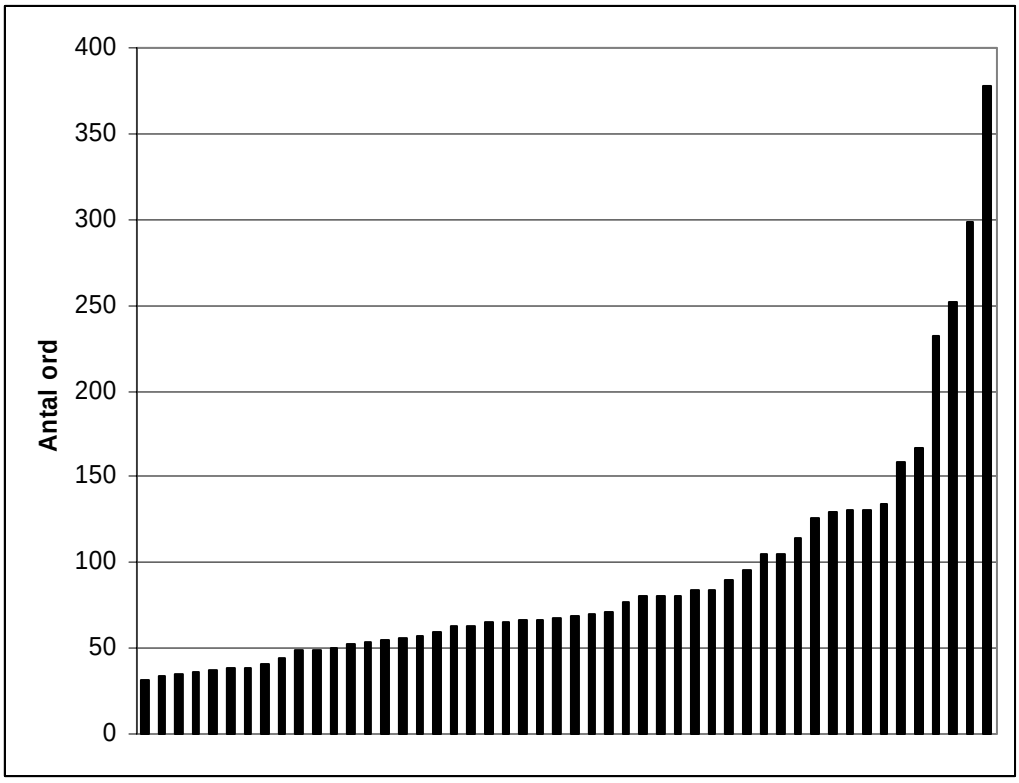
```
Skåpbilen tappade delar av sin last. <COREF ID="3">Mannen</COREF> som
färdades i <COREF ID="4">bilen</COREF> bakom tvingades väja och kolliderade
med <COREF ID="5">en lyktstolpe</COREF>. <COREF ID="6" TYPE="IDENT"
REF="4">Bilen</COREF> totalkvaddades och <COREF ID="7" TYPE="IDENT"
REF="3">mannen</COREF> fick föras till sjukhus. Olyckan inträffade vid
strax efter nio i <COREF ID="10">korsningen Drottninggatan /
Hornsgatan</COREF> i Malmö. Efter kollisionen med <COREF ID="13"
TYPE="IDENT" REF="5">lyktstolpen</COREF> fattade <COREF ID="14"
TYPE="IDENT" REF="6"><COREF ID="15" TYPE="IDENT" REF="7">mannens</COREF>
bil</COREF> eld. Branden kunde dock snabbt släckas när brandkåren kom till
<COREF ID="19" TYPE="IDENT" REF="10">platsen</COREF>. <COREF ID="20"
TYPE="IDENT" REF="15">Mannen</COREF> skadades allvarligt men dock inte
livshotande.
```

Fyra koreferenskedjor kan identifieras i dokumentet:
Mannen(3) – mannen(7) – mannens(15) – Mannen(20)
bilen(4) – Bilen(6) – mannens bil(14)
en lyktstolpe(5) – lyktstolpe(13)
korsningen Drottninggatan / Hornsgatan(10) – platsen(19)

Figur 4.1 Exempel (Tagesson 2002) på uppmärkta koreferenskedjor i ett XML-dokument.

Korpus

Två mängder av uppmärkta texter, korpus, har använts. I den första mängden, den *Carsim-genererade mängden*, är XML-dokumenten med nominalfraser och namngivna entiteter uppmärkta med hjälp av moduler i Carsim. I den andra mängden, den *manuellt uppmärkta mängden*, är nominalfraser och namngivna entiteter uppmärkta manuellt. För båda mängder är koreferenskedjor uppmärkta manuellt. De består av samma 50 texter om bilolyckor från svenska tidningar. Totalt innehåller texterna 4623 ord. Fördelningen av antal ord kan avläsas i figur 4.2. Som synes används korta texter för utvärderingen, över hälften är på under 100 ord.



Figur 4.2 Fördelning av antal ord på de 50 texterna som används för utvärderingen.

Metod

Metoden som använts för utvärderingen är *repeated hold-out* (Reich & Barai 1999). Metoden innebär att mängden med textdokument slumpvis delas in i två disjunkta mängder där den ena mängden, *träningsmängden*, används för att generera en klassificerare och utvärderingen sker på den andra mängden, *testmängden*. Proceduren upprepas under ett antal iterationer med nya slumpvisa urval tills ett stoppkriterium har mötts. I denna utvärdering har 40 slumpvis utvalda dokument använts i träningsmängden och de återstående tio i testmängden för varje iteration.

Tre olika mått har använts vid utvärderingen, *precision*, *täckning* och *F-värde*. Täckning betecknar andelen uppmärkningsbara element som har blivit uppmärkta relativt andelen som skulle ha blivit uppmärkta. Precision betecknar andelen element som är korrekt uppmärkta[*]. F-värdet är det harmoniska medelvärdet av precision och täckning. Som stoppkriterium har villkoret att F-värdet skall vara inom ±0.3 procentenheter i ett 95%-igt konfidensintervall använts.

[*] Se appendix A för en mer detaljerad definition av måtten.

Tillförlitlighet

Ett problem med repeated hold-out är att iterationerna inte är oberoende, samma databas har använts (Reich & Barai 1999). Detta innebär att stor försiktighet måste iakttas vid tolkningen av resultatet. Det finns andra utvärderingsmetoder, t.ex. *k-fold cross-validation*, som ger ett mer pålitligt resultat. Ett annat problem är att med ett slumpvis urval av texter är inte urvalet alltid representativt. Exempelvis kan några klasser vara representerade med ett fåtal, eller inga, instanser. En lösning på detta problem skulle vara *stratifiering*, att urvalet till testmängd och träningsmängd sker på ett sådant sätt att klasserna är ungefärligt likfördelade. Stratifiering har dock inte använts här.

4.2 Utvärdering av koreferensbestämning

I denna första del är samtliga tester utförda på den *Carsim-genererade mängden*, d.v.s. testerna mäter resultatet för hela systemet från en text på svenska som indata. Om inget annat nämnts har egenskapen *samma numerus* uteslutits ur testerna. Senare visas att denna egenskap är överflödig och kan ersättas med egenskapen *samma antal objekt*.

Ett första test omfattade samtliga egenskaper med komplett filtrering och egenskapsöverföring. Följande värden erhölls: täckning: 85.9%, precision: 79.5% och F-värde: 82.6%.

Filtrering

För att avgöra effekten av filtret görs ett test med vart och ett av de tolv filtervillkoren borttagna.

Filtervillkor borttaget	Resultat
Inget	**T:** 85.9% **P:** 79.5% **F:** 82.6%
Samma antal objekt = NEJ	**T:** 85.5% **P:** 77.5% **F:** 81.3%
Samma grammatiska genus = NEJ	**T:** 86.0% **P:** 78.1% **F:** 81.8%
Likhet för semantisk klass = NEJ	**T:** 86.1% **P:** 79.3% **F:** 82.5%
Samma kön = NEJ	**T:** 86.0% **P:** 79.6% **F:** 82.7%
Samma namn = NEJ	**T:** 86.3% **P:** 79.0% **F:** 82.5%
Samma ordningsnummer = NEJ	**T:** 85.7% **P:** 79.1% **F:** 82.3%
E1 är ospecificerad delmängd = JA	**T:** 86.0% **P:** 79.0% **F:** 82.3%
E2 är ospecificerad delmängd = JA	**T:** 86.2% **P:** 78.0% **F:** 81.9%
Båda räknebara = NEJ	**T:** 86.2% **P:** 76.0% **F:** 80.8%
E2 nämnd tidigare = NEJ	**T:** 87.5% **P:** 75.5% **F:** 81.1%
Samma ålderskategori = NEJ	**T:** 86.0% **P:** 78.9% **F:** 82.3%
Samma ålder = NEJ	**T:** 85.1% **P:** 80.0% **F:** 82.5%

Tabell 4.1 Test med vart och ett av filtervillkoren borttagna.

Inte alla filtervillkor har en positiv effekt men inget har heller någon signifikant negativ effekt på F-värdet. De filtervillkor som ensamma har störst effekt på F-värdet är *samma antal objekt = NEJ, båda räknebara = NEJ* och *E2 nämnd tidigare = NEJ*.

Egenskaper i egenskapsvektorn

Nästa test avser egenskaper i egenskapsvektorn med och utan filtrering. För varje test har en av egenskaperna utelämnats.

Egenskap borttagen	Använd komplett filter	Använd inget filter
Ingen	**T:**85.9% **P:**79.5% **F:**82.6%	**T:**87.1% **P:**64.8% **F:**74.3%
Avstånd	**T:**86.1% **P:**79.3% **F:**82.6%	**T:**87.0% **P:**64.4% **F:**74.0%
E1 är pronomen	**T:**86.3% **P:**79.6% **F:**82.8%	**T:**86.9% **P:**64.7% **F:**74.2%
E2 är pronomen	**T:**85.6% **P:**77.1% **F:**81.1%	**T:**86.4% **P:**62.6% **F:**72.6%
Strängmatch	**T:**86.2% **P:**79.3% **F:**82.6%	**T:**87.0% **P:**63.8% **F:**73.7%
Strängmatch för huvudord	**T:**85.8% **P:**79.9% **F:**82.8%	**T:**86.4% **P:**64.7% **F:**74.0%
Strängmatch för huvudordets lemma	**T:**82.9% **P:**80.0% **F:**81.4%	**T:**84.5% **P:**65.3% **F:**73.7%
E2 är i bestämd form	**T:**85.8% **P:**78.8% **F:**82.1%	**T:**87.6% **P:**64.8% **F:**74.5%
Likhet för semantisk klass	**T:**70.4% **P:**81.4% **F:**75.5% **T:**70.4% **P:**80.8% **F:**75.2%*	**T:**72.0% **P:**62.1% **F:**66.7%
Båda namngivna entiteter	**T:**85.7% **P:**79.1% **F:**82.3%	**T:**86.4% **P:**61.8% **F:**72.0%
Samma antal objekt	**T:**82.5% **P:**78.5% **F:**80.4% **T:**82.7% **P:**71.5% **F:**76.7%*	**T:**83.4% **P:**56.6% **F:**67.4%
Samma kön	**T:**85.4% **P:**78.7% **F:**81.9% **T:**84.2% **P:**76.3% **F:**80.1%*	**T:**86.4% **P:**58.6% **F:**69.9%
Samma namn	**T:**85.7% **P:**78.9% **F:**82.2% **T:**85.7% **P:**78.4% **F:**81.9%*	**T:**87.1% **P:**61.7% **F:**72.2%
Är närmaste element	**T:**86.1% **P:**79.2% **F:**82.5%	**T:**87.0% **P:**63.2% **F:**73.2%
Samma grammatiska genus	**T:**85.7% **P:**78.6% **F:**82.0% **T:**86.1% **P:**77.9% **F:**81.8%*	**T:**86.7% **P:**61.1% **F:**71.7%
Båda räknebara	**T:**83.3% **P:**78.4% **F:**80.8% **T:**83.7% **P:**74.6% **F:**78.9%*	**T:**85.0% **P:**60.2% **F:**70.4%
E1 är ospecificerad delmängd	**T:**85.6% **P:**78.8% **F:**82.0% **T:**86.1% **P:**79.1% **F:**82.4%*	**T:**86.5% **P:**60.8% **F:**71.4%
E2 är ospecificerad delmängd	**T:**86.0% **P:**79.4% **F:**82.6% **T:**86.0% **P:**77.6% **F:**81.6%*	**T:**86.5% **P:**61.1% **F:**71.6%
E2 nämnd tidigare	**T:**85.6% **P:**78.5% **F:**81.9% **T:**87.9% **P:**76.9% **F:**82.1%*	**T:**87.2% **P:**62.8% **F:**73.0%
Samma ordningsnummer	**T:**85.9% **P:**79.0% **F:**82.3% **T:**85.7% **P:**78.9% **F:**82.2%*	**T:**86.7% **P:**60.6% **F:**71.3%

Tabell 4.2 Tester med en egenskap borttagen från egenskapsvektorn med och utan filtrering.

* Här har även filtervillkoret associerat med egenskapen tagits bort.

Ur tabellen kan man dels utläsa effekten av enskilda egenskaper och dels effekten av filtreringen. Då samtliga egenskaper använts blir effekten av filtreringen att täckningen blir något högre utan filter medan precision och F-värde sjunker markant. Detta stämmer överens med motivationen att använda filtrering, att primärt höja precisionen genom att hindra osannolik koreferens. Resultatet utan filter är resultatet av en ren maskininlärningsbaserad metod utan handkodade regler.

Den egenskap som har störst effekt på täckning och F-värde är den domänberoende egenskapen *likhet för semantisk klass*, täckningen sjunker från 85.9% till 70.4% utan denna egenskap. För precisionen har egenskapen *samma antal objekt*, om även dess associerade filtervillkor tas bort, störst effekt. Här sjunker precisionen från 79.5% till 71.5%. Även egenskapen *båda räknebara* har stor effekt på precisionen då dess associerade filtervillkor tas bort.

Ingen av egenskaperna har ensam någon signifikant negativ effekt på F-värdet. Dock finns icke signifikanta indikationer på att egenskaperna *E1 är pronomen* och *strängmatch för huvudord* har negativ effekt.

Effekt av egenskapsöverföring

Nästa test undersöker resultatet av egenskapsöverföringen, med och utan filtrering. Endast egenskapsöverföringen med samtliga egenskaper överförda har testats (tabell 4.3).

Använd egenskapsöverföring	Använd filtrering	Resultat
Ja	Ja	**T:**85.9% **P:**79.5% **F:**82.6%
Ja	Nej	**T:**87.1% **P:**64.8% **F:**74.3%
Nej	Ja	**T:**85.8% **P:**74.4% **F:**79.7%
Nej	Nej	**T:**87.0% **P:**61.2% **F:**71.9%

Tabell 4.3 Egenskapsöverföring.

Effekten av egenskapsöverföringen är stor för precisionen, från 79.5% med egenskapsöverföring till 74.4% utan egenskapsöverföring. Effekten är något mindre för F-värdet.

En jämförelse med ett annat system

Arbetet av Soon et al. (2001) har varit en utgångspunkt för det här arbetet. En direkt jämförelse av resultaten för arbetena låter sig dock inte göra, skillnaderna är för stora. Exempelvis skiljer sig språk (engelska/svenska), beslutsträdsalgoritm (C5/ID3), nominalfraser (fullständiga/utan efterställda attribut) och utvärderingsmetod (cross validation/repeated hold-out) åt. Vad som låter sig göras är däremot se effekten av tilläggen som gjorts, relativt Soon et al., i det här arbetet. Vad som avses är undersökningen i Soon et al. som gjordes på korpus från MUC-6 (1995).

I tabell 4.4 syns de tolv egenskaperna som användes i Soon et al. samt motsvarande egenskaper i detta arbete. Egenskaperna har inte en exakt motsvarighet i samtliga fall. Egenskapen för semantisk klass är mer anpassad efter domänen (trafikolyckor) i detta arbete och *samma kön* är mer omfattande än *gender agreement*. Egenskaperna *alias* i Soon et al. och *samma namn* i det här arbetet har likartad men inte exakt samma definition. Egenskapen *distance* är mer omfattande än *avstånd*. Vidare saknas motsvarigheter för egenskaperna *demonstrative NP* och *appositive* i det här arbetet.

I Soon et al. användes endast 8 av 12 egenskaper i det slutgiltiga beslutsträdet. De övriga blev borttagna ur trädet av *beskärningsalgoritmen* i C5. Av de återstående 8 egenskaperna är det endast egenskapen a*ppositive* som inte har någon motsvarighet i detta arbete. I Soon et al. ökade F-värdet från 58.0% till 60.3% på en testmängd då denna egenskap lades till.

För det här testet används sju olika egenskaper som motsvarar de i det slutliga beslutsträdet i Soon et al., förutom a*ppositive*. Egenskapsöverföring och filtrering används inte här då det inte finns någon motsvarighet till dessa konstruktioner i Soon et al. Resultatet blev: täckning: 57.9%, precision: 74.8%, F-värde: 65.3%. Utan att en direkt jämförelse mellan detta resultat och resultatet från Soon et al. är meningsfull p.g.a. tidigare nämnda skillnader kan det nämnas att resultatet som uppnåddes för Soon et al. på MUC-6 var: täckning: 58.6%, precision: 67.3%, F-värde: 62.6%.

Soon et al. egenskap	Egenskap i detta arbete
Distance	Avstånd
i-pronoun	E1 är pronomen
j-pronoun	E2 är pronomen
String Match	Strängmatch
Definite NP*	E2 är i bestämd form
Demonstrative NP*	–
Number agreement	Samma numerus
Semantic class*	Likhet för semantisk klass
Gender agreement	Samma kön
Both proper nouns*	Båda namngivna entiteter
Alias	Samma namn
Appositive	–

Tabell 4.4 Egenskaper i Soon et al. tillsammans med motsvarande egenskaper i detta arbete.

* Dessa egenskaper blev borttagna ur det slutgiltiga beslutsträdet i Soon et al. av beskärningsalgoritmen i C5

Om sedan samtliga egenskaper i detta arbete läggs till, men utan egenskapsöverföring och filtrering, blir resultatet: täckning: 87.0%, precision: 61.2%, F-värde: 71.9%. Tillägget av egenskaperna innebär att täckningen ökar från 57.9% till 87.0% medan precisionen minskar från 74.8% till 61.2%. En slutsats av detta är att tillägget av egenskaper relativt Soon et al. förbättrar täckningen kraftigt medan tillägget av filtrering och egenskapsöverföring förbättrar precisionen. För Soon et al. skulle ett tillägg av egenskapsöverföring och filtrering troligtvis ha mer marginell effekt. De egenskaper som används för dessa konstruktioner i detta arbete är i de flesta fall inte definierade i Soon et al.

En viktig förändring relativt Soon et al. är att egenskapen för samma numerus (singular/plural) har ersatts med en egenskap för samma antal objekt. Vid ett tidigare test med samtliga egenskaper utom *samma numerus* uppnåddes resultatet: täckning: 85.9%, precision: 79.5%, F-värde: 82.6%. Om man till detta lägger till egenskapen *samma numerus* fås resultatet: täckning: 85.6%, precision: 78.2%, F-värde: 81.7%, d.v.s. en försämring av resultatet. För att vidare testa skillnaden av effekten av de båda egenskaperna görs ett antal tester med filtervillkoret *samma antal objekt = NEJ* borttaget från filtret. Resultatet syns i tabell 4.5. En större förbättring av framför allt precisionen noteras då *samma antal objekt* används istället för *samma numerus*.

***Samma antal objekt* används**	***Samma numerus* används**	**Resultat**
Nej	Ja	**T:**84.6% **P:**71.7% **F:**77.6%
Ja	Nej	**T:**85.8% **P:**77.9% **F:**81.7%
Nej	Nej	**T:**82.7% **P:**71.5% **F:**76.7%
Ja	Ja	**T:**85.3% **P:**76.1% **F:**80.5%

Tabell 4.5 Effekten av egenskaperna *samma antal objekt / samma numerus*.

Lexikaliska egenskaper

I tabell 5.6 undersöks effekten av olika lexikaliska egenskaper. I Soon et al. (2001) användes endast en lexikalisk egenskap *string match* (motsvarande *strängmatch* i detta arbete). Där ansågs egenskapen vara den som hade störst positiv effekt på F-värdet och var den egenskap som var rot i beslutsträdet.

***Strängmatch* används**	***Strängmatch för huvudordet* används**	***Strängmatch för huvudordets lemma* används**	**Resultat**
Ja	Ja	Ja	**T:**85.9% **P:**79.5% **F:**82.6%
Nej	Ja	Ja	**T:**86.2% **P:**79.3% **F:**82.6%
Ja	Nej	Ja	**T:**85.8% **P:**79.9% **F:**82.8%
Ja	Ja	Nej	**T:**82.9% **P:**80.0% **F:**81.4%
Nej	Nej	Ja	**T:**86.2% **P:**78.8% **F:**82.4%
Nej	Ja	Nej	**T:**83.1% **P:**79.6% **F:**81.3%
Ja	Nej	Nej	**T:**82.9% **P:**80.5% **F:**81.7%
Nej	Nej	Nej	**T:**79.5% **P:**78.7% **F:**79.1%

Tabell 4.6 Resultat av olika kombinationer lexikaliska egenskaper.

Ur tabell 4.6 kan det utläsas att det inte är självklart att de tre lexikaliska egenskaperna tillsammans är den mest gynnsamma kombinationen lexikaliska egenskaper. Exempelvis uppnås ett marginellt högre F-värde då egenskapen *strängmatch för huvudordet* inte används.

Intuitivt kunde det verka rimligt att de lexikaliska egenskaperna skulle ha större effekt på resultatet. Med samtliga tre egenskaper borttagna sjunker F-värdet endast från 82.6% till 79.1% jämfört med då alla tre användes. Man kan även räkna egenskapen *samma namn*, som utför strängjämförelse på element innehållande namngivna entiteter, som en lexikalisk egenskap. Tas även denna bort, tillsammans med sitt associerade filtervillkor, blir resultatet täckning: 79.4%, precision: 77.7%, F-värde: 78.6%.

Tidigare har det visats att egenskapen *likhet för semantisk klass* har stor effekt på täckningen. Detta implicerar att semantisk klass för elementen blir korrekt uppmärkt i stor utsträckning. För att identifiera semantisk klass för ett uppmärkningsbart element används ordlistor med reguljära uttryck, m.a.o. används lexikalisk information om elementet. Detta innebär att egenskapen *likhet för semantisk klass* indirekt fungerar på samma sätt som de lexikaliska egenskaperna i vissa fall. Om egenskapen ersätter de lexikaliska egenskaperna funktionellt i viss utsträckning, kan detta förklara varför resultatet inte försämrades mer då inte de lexikaliska egenskaperna användes. Då även egenskapen *likhet för semantisk klass* utesluts, tillsammans med de fyra lexikaliska, blir resultatet: täckning: 42.9%, precision: 78.3%, F-värde: 55.5%. Att täckningen här minskar med över 35 procentenheter indikerar att egenskaperna är funktionellt överlappande.

Urval av egenskaper

Ingen systematisk metod har använts för att få fram en optimal egenskapsvektor. Olika kombinationer av egenskaper har testats på grundval av resultatet av tidigare tester. Det bästa resultatet uppnåddes med samtliga egenskaper utom *E1 är pronomen* och *strängmatch för huvudordet*. Resultatet då en egenskapsvektor med de återstående 18 egenskaperna användes blev: täckning: 86.8%, precision: 80.8%, F-värde: 83.7%. Detta kan jämföras med resultatet då en egenskapsvektor med samtliga egenskaper användes: täckning: 85.9%, precision: 79.5%, F-värde: 82.6%.

Genererat beslutsträd

Det slutliga beslutsträdet som genererades bestod av 443 lövnoder. Det stora antalet beror på att ingen *beskärning* av trädet har skett. Ng & Cardie (2002a)* använde, precis som här, 18 för hand utvalda egenskaper till en egenskapsvektor i deras arbete för koreferensbestämning. Men då de använde beslutsträdsalgoritmen C4.5, med beskärning, blev resultatet istället ett beslutsträd med 17 lövnoder.

En delmängd av det genererade beslutstädet kan ses i figur 4.3. Egenskapen *likhet för semantisk klass* är rotnod i beslutsträdet.

```
Likhet för semantiska klass = JA :
 ├──Samma kön = JA :
 │    ├──E2 är pronomen = JA : KOREFERENS
 │    └──E2 är pronomen = NEJ :
 │         ...
 ├──Samma kön = NEJ : INGEN KOREFERENS
 └──Samma kön = OKÄNT :
      ├──Samma antal objekt = JA :
      │    ...
      ├──Samma antal objekt = NEJ :
      │    ...
      └──Samma antal objekt = OKÄNT :
           ...
Likhet för semantiska klass = NEJ :
 ├──Avstånd = 0 : INGEN KOREFERENS
 └──Avstånd > 0 :
      ├──Samma antal objekt = JA :
      │    ...
      ├──Samma antal objekt = NEJ : INGEN KOREFERENS
      └──Samma antal objekt = OKÄNT : INGEN KOREFERENS
Likhet för semantiska klass = OKÄNT :
 ├──Strängmatch för huvudordets lemma = JA :
 │    ├──Samma antal objekt = JA : KOREFERENS
 │    ├──Samma antal objekt = NEJ : INGEN KOREFERENS
 │    └──Samma antal objekt = OKÄNT :
 │         ...
 └──Strängmatch för huvudordets lemma = NEJ :
      ├──Samma antal objekt = JA :
      │    ...
      ├──Samma antal objekt = NEJ :
      │    ...
      └──Samma antal objekt = OKÄNT :
           ...
```

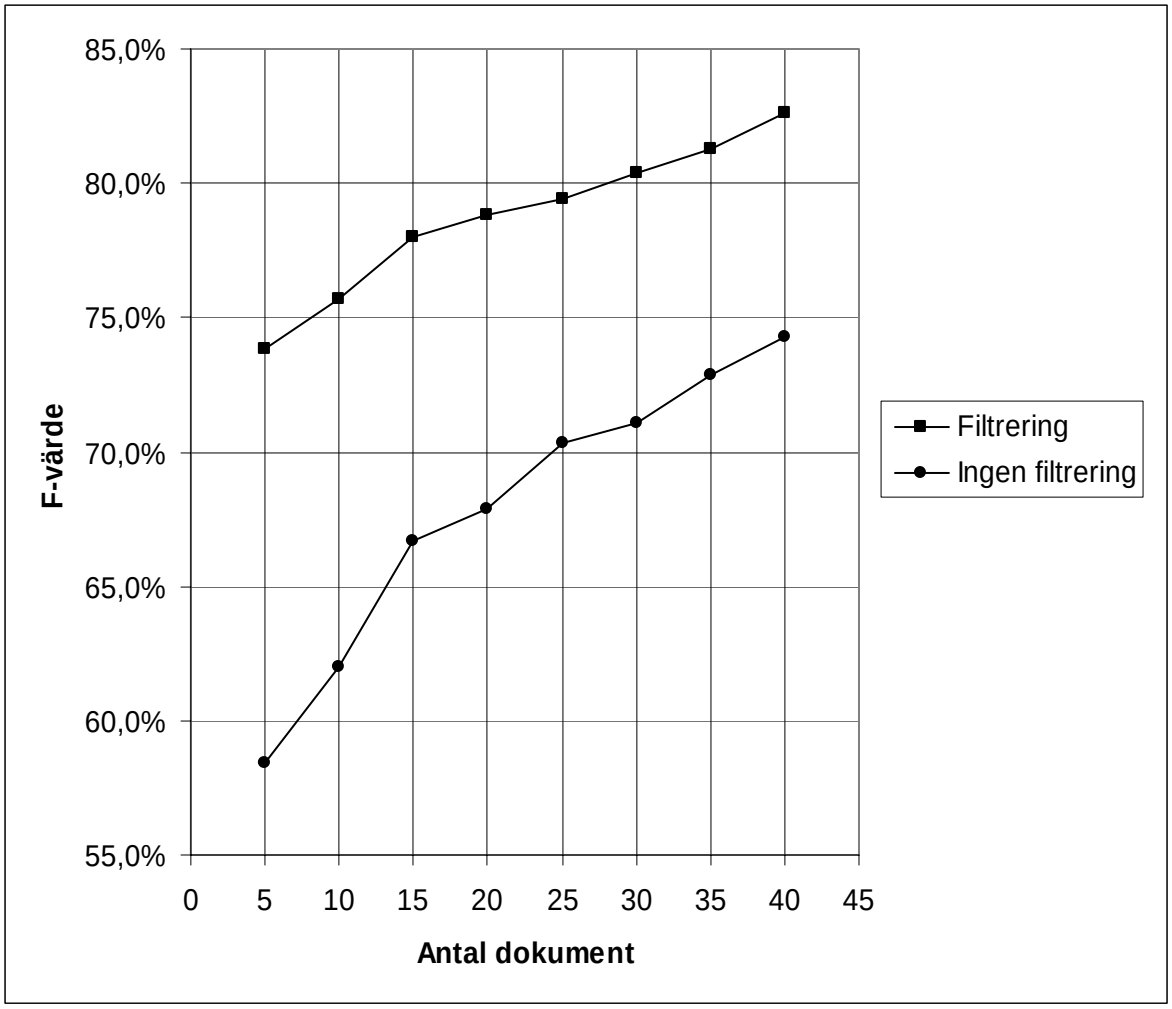
Figur 4.3 Delmängd av det slutgiltigt genererade beslutsträdet.

* Se 2.5 Relaterade arbeten för vidare beskrivning av Ng & Cardie (2002a).

Resultat med olika antal träningsdokument

Här görs tester med olika antal träningsdokument för att generera klassificeraren, med och utan filtrering. Då filtrering användes varierade resultatet från: täckning: 78.0%, precision: 69.9%, F-värde: 73.8% (5 texter) till täckning: 85.9%, precision: 79.5%, F-värde: 82.6% (40 texter). Utan filtrering varierade värdena från täckning: 80.2%, Precision: 45.9%, F-värde: 58.4% till täckning: 87.1%, precision: 64.8%, F-värde: 74.3%. Som väntat var tillväxttakten högre utan filtrering, filtret har en dämpande effekt.

I Soon et al. (2001) uppnåddes bästa resultat då träningsdokument med mellan 11000 och 17000 ord användes för att generera beslutsträdet. Då samtliga texter i både tränings- och testdokument tillsammans innehåller 4623 ord i det här arbetet är det rimligt att anta att resultatkurvan skulle ha fortsatt en bit uppåt med fler tränade dokument. Det skulle vara intressant att se hur mycket den undre kurvan skulle närma sig den övre, hur stor del av effekten filtret bidrar med som en klassificerare tränad på fler träningsexempel kunde överta.



Figur 4.4 F-värde för utvärdering med klassificerare tränad med olika antal träningsdokument, med och utan filtrering.

Manuell taggning

Texterna har gått igenom en kedja av moduler för uppmärkning av ordklasstaggar, namngivna entiteter, nominalfraser och koreferens. Resultatet kan därför sägas mäta ett komplett system som automatiskt bestämmer koreferens med en text på svenska som indata. Det kan även vara intressant att utvärdera enbart koreferensmodulen där all indata är manuellt uppmärkt. Ett test har gjorts här där modulerna för namngivna entiteter och nominalfraser inte används. Istället är denna information manuellt uppmärkt. På grund av den det stora antalet ordklasstaggar i dokumenten används modulen för ordklasstaggning i Carsim även i detta test.

Testet har utförts på samma 50 texter som i föregående test. Med den manuellt uppmärkta mängden ändrades värdet från täckning: 86.8%, precision: 80.8%, F-värde: 83.7% till täckning: 84.8%, precision: 81.8%, F-värde: 83.3%. Ett annat test gjordes både för den Carsim-genererade mängden och den manuellt uppmärkta mängden. I detta test användes samtliga texter i respektive grupp för att generera klassificeraren och varje text utvärderades för sig. Av de 50 texterna hade 41 identiska värden i de båda grupperna. Sex hade högre F-värde med den helautomatiska koreferensbestämningen och tre texter hade högre F-värde då manuellt uppmärkt indata användes. Några exempel:

I en text taggades ordet ”Hannas” som en namngiven entitet av typen NAME (namn på person). I sammanhanget framgick att det var staden ”Hannas” som avsågs. Detta ändrades manuellt och resultatet förbättrades i testet där manuellt uppmärkt indata användes.

”Han lyckades släcka branden och tillsammans med flera andra kunde han vända bilen och hjälpa personerna i *den* att komma ut”. Ordet ”den” märktes inte upp som en nominalfras i den Carsim-genererade mängden. Då detta lades till manuellt försämrades täckningen. Orsaken var att ingen koreferens kunde hittas för ”den” av klassificeraren.

I en text var frasen ”de bakomvarande” inte uppmärkt som en nominalfras. Då detta rättades till manuellt försämrades precisionen då en felaktig koreferenslänk tillkom i och med detta.

Som exemplen ovan visar kan även tillägg av korrekta nominalfraser innebära en försämring av resultatet.

Majoriteten av de ändringar som gjordes i den manuellt uppmärkta indatan relativt den automatiskt genererade var att felaktiga, enligt definitionen i detta arbete, nominalfraser togs bort. De flesta av dessa var ensamma adjektiv utan determinerare. Dessa räknas inte som uppmärkningsbara enligt definitionen och stoppades därmed av modulen för koreferensbestämning. Åtgärden att ta bort dessa från indatan hade därmed ingen effekt.

Vissa andra fraser taggades frekvent felaktigt som nominalfraser av Carsim. Exempel är väderstreck som adverb, i t.ex. ”*nordost* om Stockholm”. Ett filter med regler, i detta fall för väderstreck innan ordet *om*, för ett stoppa ett antal vanliga feltaggningar är implementerat i koreferensmodulen. Inte heller här medförde det någon skillnad då de togs bort från indatan.

En del andra fel i den automatiskt genererade indatan beskrivs i nästa avsnitt.

4.3 Fel och felkällor

Här beskrivs några fel som uppkom vid bestämning av koreferens. Även felkällor, t.ex. felaktig indata beskrivs. Exemplen som visas är hämtade från texterna som har använts i utvärderingen. Felen vid koreferensbestämningen är i form av saknade eller felaktiga länkar mellan nominalfraser. Saknade länkar orsakar täckningsfel och felaktiga länkar precisionsfel.

Felaktigt uppmärkta nominalfraser

En källa till fel vid bestämning av koreferens är felaktigt uppmärkta uppmärkningsbara element. Fel i indatan från både ordklasstaggar, namngivna entiteter och nominalfraser kan orsaka detta. Ordklasstaggaren taggar upp till 97% av alla ord korrekt (Carlberger & Kann 1999). För de typer av namngivna entiteter som märks upp är täckningen 89% och precisionen 97% för tidningsartiklar om trafikolyckor (Danielsson och Persson, 2004).

För vissa nominalfraser blev inte hela frasen uppmärkt. Orsakerna till detta varierade. För t.ex. frasen ”[E6]:an” (”[E6:an]”) orsakades detta av att frasen inte märktes upp som ett ord av ordklasstaggaren. I en del fall märktes verb felaktigt upp som framförställda attribut: ”[vållade stora problem]” (”vållade [stora problem]”). Regler saknas för att identifiera egennamn på andra språk, vilket illustreras av frasen ”[J] [Mascis and the] [Fog] spelade” (”[J Mascis and the Fog] spelade”). I ett annat exempel: ”[Aftonbladets Per Bjurman]” (”[[Aftonbladets] Per Bjurman]”) blev inte ”Aftonbladets” identifierad som en inre nominalfras. Detta berodde på att ordklasstaggaren inte märkte upp ordet som genitiv. I ett annat fall orsakade en felstavning att den inre nominalfrasen inte märktes upp: ”[socialnämnden] [ordförande Boris von Uexküll]” (”[[socialnämnden(s)] ordförande Boris von Uexküll]”).

Som beskrivits i 4.2, *manuell taggning*, filtrerades en mängd felaktiga nominalfraser bort av koreferensmodulen. Ett mindre antal återstod dock. I frasen ”stå [parkerad mitt] på” (”stå parkerad mitt på”) märktes ”mitt” felaktigt upp som ett substantiv istället för ett adverb.

Ett fåtal nominalfraser blev inte uppmärkta. Frasen ”De bakomvarande” blev inte uppmärkt som en nominalfras. ”bakomvarande” märktes upp som ett adjektiv i participform (verbavlett adjektiv) av ordklasstaggaren. Ingen regel fanns i modulen för nominalfrasbestämning för fraser med particip som huvudord.

Felaktig koreferens

Förutom felaktigt uppmärkta nominalfraser finns en rad andra orsaker till felaktiga eller saknade koreferenslänkar. I ett par fall berodde detta på elementattributen bestämdes felaktigt. En vanligare orsak var att filtret stoppade korrekt koreferens. Filtervillkoret *E2 nämnd tidigare = NEJ* stoppade koreferens där anaforen inleddes med ordet ”en”, t.ex. ”en bil”. I de allra flesta fall stoppades koreferens korrekt, det är vanlig konstruktion då ett objekt introduceras i en text. Utvärderingen gjordes på tidningsartiklar där det ofta förekommer en inledande del. Objektet introduceras i de fallen två gånger. Detta medför att filtervillkoret ibland förhindrar koreferens i texter med en inledande sammanfattning. Ett annat filtervillkor som kan orsaka att koreferens inte kan bestämmas är *samma grammatiska genus = NEJ*. Detta sker i fall där antecedent och anafor har olika grammatiska genus: ”huset” (neutrum) och ”byggnaden” (utrum).

Den vanligaste orsaken till att koreferens inte bestäms korrekt är att egenskaperna i egenskapsvektorn inte innehåller tillräckligt med information för att skapa korrekta regler. Detta kan bero på att den nödvändiga informationen finns i texten mellan nominalfraserna eller att den finns utanför texten, d.v.s. underförstådd kunskap om världen.

Olyckan inträffade när **en bil** på väg nerför backen i riktning mot Jönköping gjorde en tvär omkörning. Föraren i **den framförvarande bilen**, **en Mazda**, tvingades göra en häftig undanmanöver och kom ut i mötande körfält. I **bilen** fanns tre personer. **Bilen** kolliderade med en Peugeot, med tre personer i. **Den omkörande bilen** försvann från olycksplatsen. Sent i går kväll hade polisen inga spår efter **den bilen** som smet.

Korrekta koreferenskedjor.

Olyckan inträffade när **en bil** på väg nerför backen i riktning mot Jönköping gjorde en tvär omkörning. Föraren i **den framförvarande bilen**, **en Mazda**, tvingades göra en häftig undanmanöver och kom ut i mötande körfält. I **bilen** fanns tre personer. **Bilen** kolliderade med **en Peugeot**, med tre personer i. **Den omkörande bilen** försvann från olycksplatsen. Sent i går kväll hade polisen inga spår efter **den bilen** som smet.

Genererade koreferenskedjor.

Figur 4.5 Exempeltext (Hansson 2000) med korrekta och genererade koreferenskedjor..

I figur 4.5 illustreras svårigheterna att bestämma koreferens korrekt. Det är uppenbart att egenskaperna i egenskapsvektorn är otillräckliga för att skapa regler som med säkerhet kan bestämma koreferens i exemplet. Nominalfrasen ”tre personer” förekommer på två ställen. Trots att strängrepresentationen är identisk och de har samma semantiska klass och samma antal objekt så koreferera inte de två förekomsterna av nominalfrasen. För att avgöra koreferens för objekten i texten behövs mycket av informationen som finns i texten mellan nominalfraserna. Verbfraser måste tolkas och kopplas till nominalfraser. Denna information finns inte representerad i egenskaperna i egenskapsvektorn.

Nominalfrasen *den*

En till synes allvarlig brist är att beslutsträdet inte har någon regel som tillåter nominalfrasen ”den” att koreferera. Dock påverkar detta inte utvärderingsresultatet i någon större omfattning. I de 50 texterna förekommer endast 9 ”den” i någon koreferenskedja. I de allra flesta fall ingår istället ”den” som determinerare i en nominalfras.

Indata till beslutsträdsalgoritmen ID3 är en mängd egenskapsvektorer utvunna från antecedent-anaforpar. De är klassificerade som koreferera nde eller icke-koreferera nde. En absolut förutsättning för att en positiv lövnod (koreferens) skall genereras är att det för en viss egenskapsvektor finns en majoritet positiva exempel.

”... inträffade när **en personbil** skulle göra en vänstersväng och en lastbil som kom från mötande håll körde in i **den**.”

I exemplet ovan genereras det positiva exemplet ”en personbil – den” och de negativa exemplen ”en vänstersväng – den”, ” en lastbil – den” och ”mötande håll – den”. För tre av de fyra exemplen, för alla utom ”mötande håll – den”, genereras identiska egenskapsvektorer. Detta innebär att för denna egenskapsvektor är de negativa exemplen i majoritet och ingen regel för koreferens kan bestämmas av beslutsträdsalgoritmen.

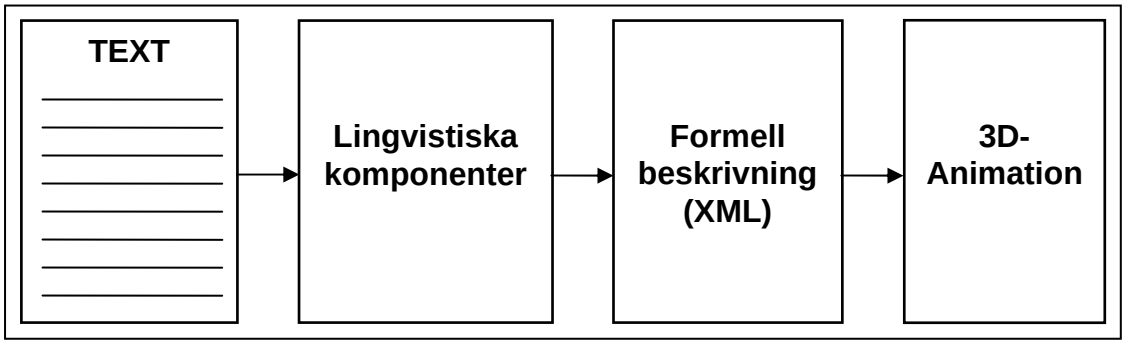
Detsamma har gällt för samtliga träningsexempel där ”den” har varit en del av det koreferera nde paret. Inga regler för koreferens har kunnat bestämmas då fler negativa än positiva träningsexempel har genererats för varje egenskapsvektor där ”den” varit ett element.

5 Integration i Carsim

I detta kapitel ges en kort sammanfattning av funktionen hos Carsim samt en beskrivning av hur koreferensbestämningen i det här arbetet används för att bestämma refererande nominalfraser i Carsim.

5.1 Carsim

Carsim (Johansson et al. 2004; Dupuy et al. 2001) är ett system som genererar en tredimensionell simulering av en trafikolycka utifrån en text som beskriver olyckan. Information i texten utvinns med hjälp av lingvistiska moduler. Den information som är relevant för simuleringen sammanställs i en formell beskrivning med objekt och händelser som används för att skapa simuleringen.



Figur 5.1 Schematisk översikt över modulerna i Carsim.

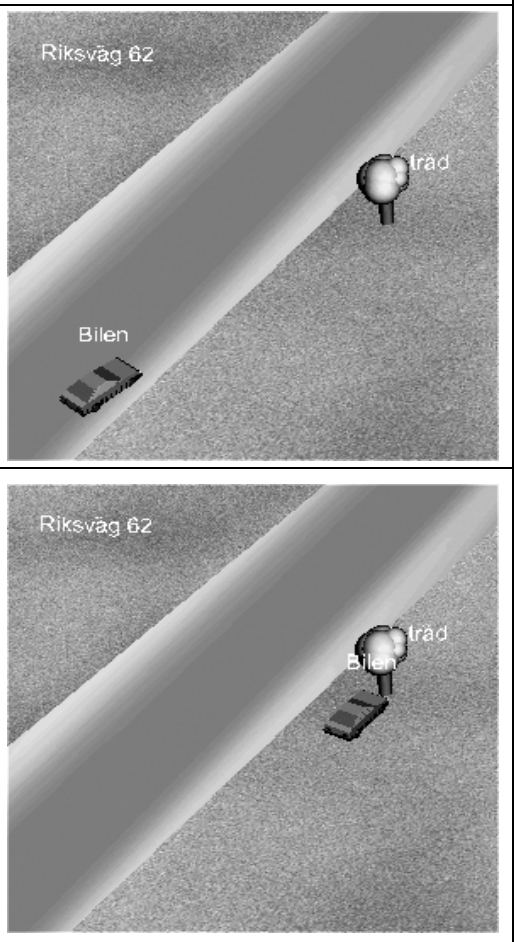
De lingvistiska komponenterna består av en pipeline av moduler som extraherar olika typer av information från texten. Det finns moduler som tokeniserar texten, hittar meningar, identifierar ordklasstaggar, hittar satsgränser, identifierar egennamn, identifierar nominalfraser, identifierar tidsuttryck och identifierar händelser. Utifrån denna information skapas en formell beskrivning, i XML-format, av olyckan som sedan används för att skapa 3D-simuleringen. Den formella beskrivningen består av tre typer av objekt:

- Ett *scenobjekt*, vilket beskriver de statiska parametrarna av omgivningen, t.ex. väder, vägförhållanden och vilken typ av väg olyckan utspelar sig på.
- Ett eller flera *vägobjekt*, t.ex. bilar, motorcyklar och träd samt deras associerade rörelsescheman.
- En eller flera *händelser* med de inblandade vägobjekten.

En 23-årig man dödades och två personer båda födda 1982, skadades allvarligt vid en singelolycka på Riksväg 62 söder om Sysslebäck sent i går kväll. Bilen gick av okänd anledning av vägen och rände in i ett träd. (Sveriges radio 2002)

Template

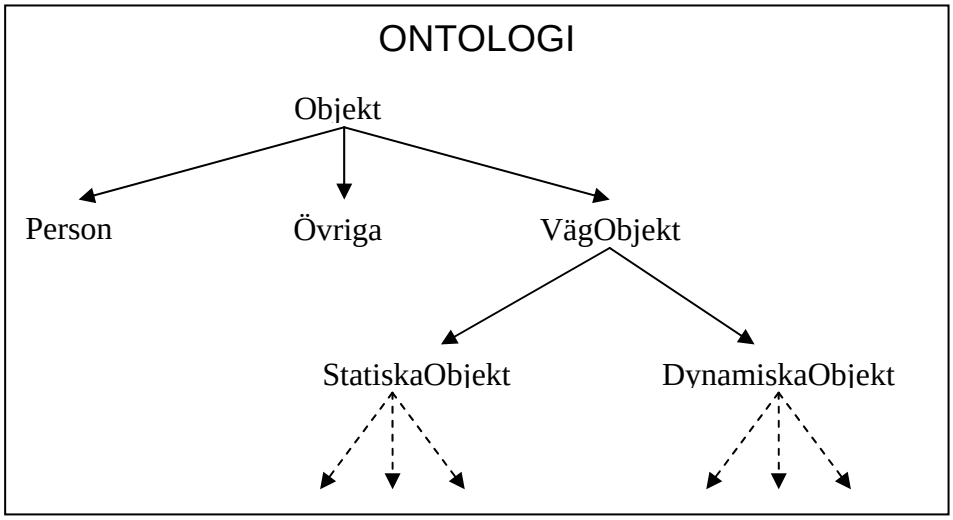
Scene	Scene	Location	söder om Sysslebäck		
		RoadConfiguration	StraightRoad	RoadName	Riksväg 62
				LeftAttrs	SideAttrs
				RightAttrs	SideAttrs
				Id	RoadConfig56
		Environment	rural		
Objects	Car	Id	RoadObject40		
		IntroducedAs	Bilen		
		Positions			
		Directions			
	Tree	Id	RoadObject41		
		IntroducedAs	ett träd		
		Positions			
Events	LeaveRoad	Id	Event34		
		Actor	(RoadObject40)		
		Positions			
		Directions			
		Times			
	Impact	Id	Event35		
		Actor	(RoadObject40)		
		Victim	(RoadObject41)		
		Positions			
		Directions			
		Times	StrictlyAfter	RelativeTo	(Event34)



Figur 5.2 Exempel på formell beskrivning av en text med tillhörande scenobjekt, vägobjekt och händelser samt den av Carsim genererade 3D-simuleringen.

5.2 Implementering av referensbestämning

I det här arbetet identifieras vägobjekt och andra objekt vars typ finns definierad i en *ontologi* i Carsim. Begreppet ontologi skapades ursprungligen av Aristoteles och betecknar en klassificering av uttryck efter deras betydelse. Ontologin i Carsim är uppbyggd på samma sätt som den semantiska klasshierarkin i koreferensmodulen, i en hierarkisk trädstruktur. Det finns sammanlagt 57 olika typer av objekt i ontologin varav alla utom två representerar vägobjekt (se figur 5.3). Dessa två är *person* för personer som ej syns i simuleringen, t.ex. bilförare, samt *övriga*. Vägobjekten är indelade i två huvudklasser, *statiska* och *dynamiska* objekt. Exempel på statiska objekttyper är träd, stenar, trafikljus och snö på vägbanan och exempel på dynamiska objekttyper är olika typer av fordon, husvagnar och personer direkt inblandade i en olycka.



Figur 5.3 Delmängd av ontologin som används i Carsim.

För varje referent, representerad av en koreferenskedja eller enskild nominalfras, skall det avgöras om dess typ finns definierad i ontologin. Ontologitypen identifieras på två olika sätt. Är den semantiska klassen definierad för referenten kan i vissa fall denna översättas direkt till en ontologityp. I annat fall görs strängjämförelser med reguljära uttryck från en databas för att avgöra om en nominalfras refererar till ett ontologiobjekt.

I figur 5.4 visas exempel på en text med funna koreferenskedjor samt med funna ontologireferenser. Tre av de fyra koreferenskedjorna samt en av de enskilda nominalfraserna har en motsvarighet i ontologin. Av dessa fyra är tre vägobjekt och kommer alltså att användas i simuleringen.

Koreferens	**Referenser till ontologin**
Skåpbilen tappade delar av sin last. Mannen som färdades i bilen bakom tvingades väja och kolliderade med en lyktstolpe. Bilen totalkvaddades och mannen fick föras till sjukhus. Olyckan inträffade vid strax efter nio i korsningen Drottninggatan / Hornsgatan i Malmö. Efter kollisionen med lyktstolpen fattade mannens bil eld. Branden kunde dock snabbt släckas när brandkåren kom till platsen. Mannen skadades allvarligt men dock inte livshotande.	Skåpbilen tappade delar av sin last. Mannen som färdades i bilen bakom tvingades väja och kolliderade med en lyktstolpe. Bilen totalkvaddades och mannen fick föras till sjukhus. Olyckan inträffade vid strax efter nio i korsningen Drottninggatan / Hornsgatan i Malmö. Efter kollisionen med lyktstolpen fattade mannens bil eld. Branden kunde dock snabbt släckas när brandkåren kom till platsen. Mannen skadades allvarligt men dock inte livshotande.
Ontologireferenser Skåpbilen → Objekt/VägObjekt/DynamisktObjekt/Fordon/Van bilen – Bilen – mannens bil → Objekt/VägObjekt/DynamisktObjekt/Fordon/Personbil Mannen – mannen – mannens – Mannen → Objekt/Person en lyktstolpe – lyktstolpen → Objekt/VägObjekt/StatisktObjekt/Hinder/VertikaltHinder/Lyktstolpe	

Figur 5.4 En text (Tagesson 2002) med koreferenskedjor och ontologireferenser.

5.3 Utvärdering

Carsim har sedan tidigare en modul för att identifiera ontologireferenser. Koreferensdelen i denna modul använder en enkel algoritm, baserad på Appelt & Israel (1999), som för varje nominalfras i bestämd form antar koreferens med närmast tidigare nominalfras om de är konsistenta enligt ontologin. Nominalfraser i obestämd form antas vara referenser till tidigare icke nämnda objekt. För både koreferenskedjor och icke korefererande nominalfraser används strängjämförelser med reguljära uttryck för att avgöra referens till ett ontologiobjekt.

Ingen omfattande utvärdering har genomförts för referensbestämningen. Ett urval gjordes på tolv korta texter med sammanlagt 86 referenser till ontologiobjekt. De korrekta referenserna uppmärktes manuellt och en jämförelse, med hjälp av algoritmen beskriven i appendix A.2, gjordes med resultatet från både den gamla referensbestämningen i Carsim och med referensbestämningen i det här arbetet. Resultaten som uppnåddes var med den gamla referensbestämningen: täckning: 68%, precision: 82% och F-värde: 75%. Med referensbestämningen i det här arbetet uppnåddes resultatet: täckning: 90%, precision: 85% och F-värde: 88%. På grund av det begränsade urvalet av texter för utvärderingen är den enda slutsats man kan dra är det finns starka indikationer på en förbättring av resultatet på korta texter om trafikolyckor med den nya metoden för referensbestämning relativt den gamla.

6 Framtida arbete

I det här kapitlet ges förslag på möjliga utökningar och förbättringar av koreferensbestämningen i det här arbetet.

6.1 Möjliga förbättringar

Större träningsmängd

Som konstaterades i kapitel 4 finns det indikationer på att bättre resultat hade kunnat uppnås med fler manuellt uppmärkta texter att träna klassificeraren på. I Soon et al. (2001) uppnåddes bästa resultat då träningsdokument med mellan 11000 och 17000 ord användes för att generera en klassificerare. I detta arbete innehåller samtliga texter i både tränings- och testdokument tillsammans 4623 ord.

Utökning och förbättring av egenskaper

För att få en bra klassificerare är en grundförutsättning att ”rätt” egenskaper finns i egenskapsvektor och att deras värden är så korrekt identifierade som möjligt. Det finns mängder av egenskaper som potentiellt kunde förbättra klassificeraren i det här arbetet som inte finns implementerade. I t.ex. Ng & Cardie (2002a) används sammanlagt 53 olika egenskaper mot 20 i det här arbetet. Visserligen användes inte samtliga dessa i det slutgiltiga beslutsträdet med för hand utvalda egenskaper men motsvarigheten till ett antal av dessa egenskaper skulle ha potential att förbättra resultatet i det här arbetet. Exempelvis visade de att en utökning av antalet lexikaliska (strängjämförande) egenskaper förbättrade resultatet.

Filter och egenskapsöverföring

Av de arbeten om koreferensbestämning som har undersökts inför detta arbete har inga hittats som använder en kombination av beslutsträd och filtrering på samma sätt som i det här arbetet. Inte heller har någon konstruktion liknande egenskapsöverföring vid klustring funnits. I detta arbete har det visats att dessa tillägg har bidragit till en stor förbättring av resultatet. Därför skulle det vara principiellt intressant att undersöka dessa vidare. Urvalet av egenskaper för filter och egenskapsöverföring har här skett på ”måfå”, med lingvistisk intuition, utan någon systematisk urvalsmetod.

Det är möjligt att en del av förbättringen med dessa konstruktioner beror på specifika egenheter hos systemet i det här arbetet. Det är rimligt att anta att effekten skulle bli mindre på ett system utvecklat för allmänna, snarare än domänspecifika, texter. Detta kommer sig av egenskapen för semantisk klass i detta fall skulle bli svårare att identifiera korrekt. Utvärderingen i kapitel 4 för olika antal träningsdokument indikerar att skillnaden med och utan filtrering minskar ju fler träningstexter som används för att generera klassificeraren. Det är osäkert hur stor effekten filtret kommer att ha med en klassificerare tränad på optimalt antal träningsinstanser.

Beslutsträdsalgoritm

ID3 (Quinlan 1993) har använts som beslutsträdsalgoritm i det här arbetet. Dess efterföljare C4.5 (Quinlan 1993) och den kommersiellt utvecklade C5 (Rulequest Research 2004) har ett antal egenskaper som ID3 saknar. Exempelvis använder de *beskärning* som tar bort grenar från beslutsträdet som försämrar resultatet. De använder också ett mått, g*ain ratio,* som ersätter delningskriteriet *information gain* i ID3 och innebär bättre klassificering för attribut med många värden. Att ersätta ID3 med en mer utvecklad beslutsträdsalgoritm borde medföra en bättre klassificerare.

Andra maskininlärningsbaserade metoder

Det finns andra maskininlärningsbaserade metoder, som inte bygger på beslutsträd, som kan vara värda att utforska. *Support vector machines* (SVM) (Vapnik 1998) är en sådan metod.

Klustringsalgoritm

I detta arbete används en klustringsalgoritm som skapar ett korefererande par med anaforen och den närmaste antecedenten. I Ng & Cardie (2002a) används istället en s.k. *bäst-först* klustringsalgoritm. Detta innebär att, för varje anafor, den potentiella antecedent som av klassificeraren bedömts ha högst sannolikhet för koreferens (om någon >50%) bildar ett korefererande par med anaforen. Här är även urvalsmetoden för vilka träningsexempel som klassificeraren tränas på anpassad efter klustringsalgoritmen. Denna förändring innebar, jämfört med motsvarande klustringsalgoritm som använts i det här arbetet, för Ng & Cardie (2002a) en förbättring av resultatet (Ng 2002).

Bestämning av anaforer

Detta motsvarar steg 3 i den generiska algoritmen i 2.4 (sid. 8) och är inte implementerat i detta arbete. I Ng & Cardie (2002c) presenteras en metod för att bestämma om en nominalfras är en möjlig anafor eller inte. Syftet med att bestämma vilka nominalfraser som är anaforiska är dels att spara processortid, klustringsalgoritmen behöver inte söka efter icke-anaforiska nominalfraser, och dels för öka precisionen. Varje antecedent som identifierats för en icke-anaforisk nominalfras är uppenbarligen felaktig och skadar därför precisionsvärdet i koreferenssystemet. Metoden de använder är maskininlärningsbaserad med beslutsträdsalgoritmen C4.5 tillsammans med ett par handkodade regler. Resultatet blir en signifikant förbättring av precision och F-värde och en försämring av täckningen i ett koreferenssystem där bestämning av anaforer finns implementerat.

Skillnaden mellan bestämning av anaforer och filtreringen i det här arbetet är att filtreringen förhindrar koreferens mellan specifika antecedent-anaforpar under klustringen medan ”anaforfiltret” förhindrar enskilda element att fungera som anaforer i någon koreferensrelation.

6.2 Möjliga utökningar

Bestämning av ordklasstaggar, namngivna entiteter och nominalfraser

Modulerna för bestämning av ordklasstaggar, namngivna entiteter och nominalfraser som används för koreferensbestämningen har inte varit en del av detta arbete utan en integrerad del i Carsim. En möjlig utvidgning av detta arbete skulle vara att skapa ett komplett fristående system för koreferensbestämning med motsvarande moduler integrerade.

I framförallt modulerna för namngivna entiteter och nominalfraser finns möjligheter till förbättringar och utökningar. Modulen för namngivna entiteter är domänberoende för texter om trafikolyckor och taggar endast nio olika typer av namngivna entiteter. Nominalfraserna som taggas är inte fullständiga, de saknar efterställda attribut och inre nominalfraser. Att skapa ett system för koreferensbestämning av fullständiga nominalfraser är en möjlig utvidgning. Att korrekt detektera fullständiga svenska nominalfraser är dock ansett som ett mycket svårt problem (Johansson 2000). Dels är strukturen hos svenska nominalfraser invecklad och dels, liksom för koreferensbestämning, krävs i många fall ingående kunskap om världen. Detta är orsaken till att detektering av fullständiga nominalfraser inte finns implementerat i Carsim.

Andra typer av relationer

En annan möjlig utökning av systemet är att bestämma andra typer av relationer än identitetsrelationen mellan nominalfraserna. Det finns ett stort antal möjliga typer av relationer mellan nominalfraser. Ett par möjliga relationer att undersöka är: *Mängd/delmängd-relation* – bilen i nominalfrasen *en bil* kan vara en delmängd av bilarna i nominalfrasen *tre bilar*. *Helhet/del-relation* – objektet som avses i nominalfrasen *ett hjul* kan vara en del i *bilen*.

7 Slutsatser

7.1 Arbetet

Jag har utvecklat ett program i Carsim för att bestämma koreferens för nominalfraser. Programmet är utvecklat för svenska texter där nominalfraser, egennamn och ordklasstaggar är uppmärkta i indatan. Nominalfraserna är inte fullständiga, efterställda attribut saknas och endast en typ av inre nominalfraser finns definierad. Så vitt jag vet är detta det första helautomatiska systemet för koreferensbestämning avsett för svenska texter. Vidare har koreferensmodulen integrerats i Carsim för att detektera vägobjekt. Detta görs genom att finna referenser från nominalfraser till objekttyper definierade i en ontologi.

Metoden för koreferensbestämning är baserad på ett arbete av Soon et al. (2001). Beslutsträdsalgoritmen ID3 har använts. I tillägg till arbetet av Soon har ett antal egenskaper lagts till, både domänoberoende och domänberoende. Totalt används 20 lexikaliska, grammatiska, positionella och semantiska egenskaper. Jag har även utökat Soons arbete med filter och egenskapsöverföring. Filtret består av för hand skapade regler och stoppar antecedent-anaforpar där koreferens är osannolik. Egenskapsöverföringen används under exekveringen då koreferenskedjorna bestäms. Varje nominalfras i en kedja får det mest specifika värdet som någon nominalfras i kedjan har, för vissa egenskaper. Dessa egenskaper är primärt semantiska.

7.2 Resultat

Koreferensmodulen har utvärderats på texter om trafikolyckor. Bäst resultat uppnåddes med ett beslutsträd där 18 av de 20 egenskaperna användes. Dessa valdes för hand. Resultatet som uppnåddes här var: täckning: 86.8%, precision: 80.8%, F-värde: 83.7%. Resultatet är markant bättre än vad som uppnåtts för något system utvärderat på MUC-6 (1995) och MUC-7 (1997). Resultaten är dock inte direkt jämförbara. Framförallt avser koreferensbestämningen i MUC-6 och MUC-7 fullständiga nominalfraser. Texterna som använts i det här arbetet är korta, enkelt uppbyggda och de är många fall likartade. Utvärderingen skett på en betydligt mindre textmassa och med en mindre pålitlig utvärderingsmetod än i t.ex. Soon et al.

Egenskapen *likhet för semantisk klass* hade anpassats för texter om trafikolyckor och hade stor effekt på täckningen. Motsvarande egenskap i Soon et al. var domänoberoende och hade mycket mindre effekt. Egenskapen *samma antal objekt*, som inte har någon motsvarighet i Soon et al., hade stor effekt på precisionen. Detsamma gäller för filtret och egenskapsöverföringen.

En mindre utvärdering gjordes på den i Carsim integrerade modulen för bestämning av referenser till ontologiobjekt. Sannolikt förbättrades resultatet jämfört med en tidigare metod som användes.

7.3 Fortsatta undersökningar

Resultatet från andra arbeten indikerar att resultatet kan förbättras signifikant. Andra algoritmer för att skapa en klassificerare, t.ex. beslutsträdsalgoritmen C4.5 har stor potential att förbättra resultatet. En klassificerare tränad på fler texter kan också innebära förbättringar. Fler egenskaper, annan klustringsmetod och bestämning av anaforer är andra möjliga vägar för att skapa ett bättre system för koreferensbestämning.

Referenser

Appelt, Douglas E & Israel, David (1999). Introduction to information extraction technology. *Tutorial Prepared for IJCAI-99*. Artificial Intelligence Center, SRI International.

Azzam, Saliha, Humphreys, Kevin & Gaizauskas, Robert (1999). Using coreference chains for text summarization. I *Proceedings of the Workshop on Coreference and Its Applications*.

Baldwin, Breck, Morton, Tom, Bagga, Amit, Baldridge, Jason, Chandraseker, Raman, Dimitriadis, Alexis, Snyder, Kieran & Wolska, Magdalena (1997). Description of the upenn camp system as used for coreference. I *Proceedings of the 7th Message Understanding Conference* (MUC-7).

Cardie, Claire & Howe, Nicholas (1997). Improving minority class prediction using case-specific feature weights. I *Proceedings of the Fourteenth International Conference on Machine Learning*, ss. 57-65.

Carlberger, Johan & Kann, Viggo (1999). Implementing an efficient part-of-speech tagger. *Software Practice and Experience*, 29, ss. 815-832.

Danielsson, Magnus & Persson, Lisa (2004). *Name extraction in car accident reports for Swedish*. Teknisk rapport, LTH, Department of Computer science. Januari.

Dupuy, Sylvain, Egges, Arjan, Legendre, Vincent & Nugues, Pierre (2001). Generating a 3D simulation of a car accident from a written description in natural language: The Carsim system. I *Proceedings of The Workshop on Temporal and Spatial Information Processing*, ss. 1–8, Toulouse, 7 juli. Association for Computational Linguistics.

Frege, Gottlob. (1892). Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*. vol. 100, ss. 25–50.

Hansson, Anita (2000). Sex skadades i olycka utanför Jönköping. *Aftonbladet*, 30 september.

Hultman, Tor G. (2003). *Svenska akademiens språklära*. Stockholm: Svenska Akademien.

Johansson, Richard, Williams, David, Berglund, Anders & Nugues, Pierre (2004). Carsim: A System to Visualize Written Road Accident Reports as Animated 3D Scenes. I *Proceedings of the Second Workshop on Text Meaning and Interpretation, 42nd Annual Meeting of the Association of Computational Linguistics*, ss. 57–64, Barcelona, Spain. Association for Computational Linguistics.

Johansson, Victoria (2000). *NP-detektion. Utvärdering och förslag till förbättringar av Granskas NP-regler*. Stockholms universitet, Institutionen för Lingvistik.

McCarthy, Joseph F. & Lehnert, Wendy G. (1995). Using decision trees for coreference resolution. I *Proceedings of the Fourteenth International Joint Conference on Artifical Intelligence*, ss. 1050-1055.

MUC-6 (1995). MUC-6 Coreference task definition. I *Proceedings of the Sixth Message Understanding Conference*. v2.3, 8 september.

MUC-7 (1997). MUC-7 Coreference task definition. I *Proceedings of the Seventh Message Understanding Conference*. v3.0, 13 juli.

Ng, Vincent & Cardie, Claire (2002a). Improving machine learning approaches to coreference resolution. I *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

Ng, Vincent & Cardie, Claire (2002b). Combining sample selection and error-driving for machine learning of coreference rules. I *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, ss. 55-62.

Ng, Vincent & Cardie, Claire (2002c). Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. I *19th International Conference on Computational Linguistics*.

Ng, Vincent (2002). *Machine learning for coreference resolution: Recent successes and future challenges*.

Pantel, Patrick André (2003). Clustering by Committee. *University of Alberta, Department of Computing Science*.

Quinlan, John Ross (1993) *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA.

Reich, Yoram & Barai, S. V. (1999). Evaluating machine learning models for engineering problems. I *Artificial Intelligence in Engineering*, vol 13, ss. 257-272.

Rulequest Research (2004). *Data Mining Tools See5 and C5.0*. (Elektronisk). Tillgänglig: <http://www.rulequest.com/see5-info.html>. (November 2004).

Shannon, Claude. 1948. A Mathematical Theory of Communication. I *Bell System Technology Journal*, vol 27, ss. 379-423, 623-56.

Soon, Wee Meng., Ng, Hwee Tou. & Lim, Daniel Chung Yong (2001). A machine learning approach to coreference resolution of noun phrases. I *Computational lingustics*, vol. 27: 4, ss. 521-544.

Tagesson, Eric (2002). Bilolycka mitt i morgonrusningen. *Sydsvenskan*, 18 juli.

van Deemter, Kees & Kibble, Rodger (2000). On coreferring: Coreference in MUC and related annotation schemes. I *Computational Linguistics*, vol. 26: 4, ss.629–637.

Vapnik, Vladimir (1998). *Statistical learning theory*. New York: Wiley.

Vilain, Marc, Burger, John, Aberdeen, John, Connolly, Dennis & Hirschman, Lynette (1995). A model-theoretic coreference scoring scheme. I *Proceedings of the 6th Message Understanding Conference* (MUC-6), ss. 45–52, San Mateo, Cal.: Morgan Kaufmann.

A Utvärderingsmetoder

I detta appendix visas metoder för att utvärdera resultatet av koreferens- och referensbestämningen. Den första utvärderingsmetoden (Vilain 1995 se Baldwin et al. 1997) används även för att utvärdera koreferens i MUC-6 (1995) och MUC-7 (1997). Metoden utgår från länkarna i koreferenskedjorna. Den kan dock inte användas för att utvärdera referensbestämningen då den inte tar hänsyn till enskilda referenser som inte ingår i en koreferenskedja (Baldwin et al. 1997). Den andra metoden (Baldwin et al. 1997) är mer lämplig för referensbestämning då utvärderingen görs utifrån enskilda element.

A.1 Utvärdering av koreferens

För utvärderingen används två mängder med koreferenskedjor, *key* och *response*, som definieras enligt följande:

Key avser de koreferenskedjor som är manuellt uppmärkta i dokumentet, sanningen.
Response avser de koreferenskedjor som är maskinellt uppmärkta i dokumentet av systemet för koreferensbestämning.

Key, sanningskedjor.
1←2←3←4 5←6 7←8←9 A←B←C

Response, maskinellt genererade kedjor.
1←2←3←4 7←8 9←——B C←D E←F←G←H

Figur A.1 Exempel på *key* och *response*. *Key* består av fyra koreferenskedjor och *response* av fem. De har fyra länkar gemensamt.

Vi låter $key = [K_1, K_2, \ldots, K_n]$ beteckna mängden av de korrekt uppmärkta koreferenskedjorna där varje $K_i \in key$ är en ekvivalensklass för koreferensrelationen. $Response = [R_1, R_2, \ldots, R_m]$ betecknar de maskinellt uppmärkta koreferenskedjorna. *Key* är en partition på mängden av refererande uttryck då elementen är parvis disjunkta och täcker mängden. På samma sätt är *response* en partition. För att avgöra hur god överensstämmelse som finns mellan *key* och *response* måste någon form av jämförelse göras som genererar numeriska värden som ett mått på samstämmigheten. Ett sätt att göra detta är att utgå från länkarna mellan elementen i koreferenskedjorna. Vi definierar två mått på överensstämmelsen mellan *key* och *response*:

$$\text{täckning} = \frac{\text{antalet korrekta länkar i } response}{\text{antalet länkar } key}$$

$$\text{precision} = \frac{\text{antalet korrekta länkar i } response}{\text{antalet länkar } response}$$

För varje ekvivalensklass i *key* respektive *response* är det uppenbart att antalet länkar är ett mindre än antal element. Antalet korrekta länkar i *response* kan även uttryckas som antalet länkar gemensamma för både *key* och *response*. För att räkna ut antalet gemensamma länkar kan man först för varje K_i skapa en partition med avseende på elementen i *response*. Partitionen får egenskapen att dess kardinalitet är lika med antalet saknade länkar i *response* relativt K_i plus ett. Antalet gemensamma länkar kan sedan uttryckas som differensen av antalet länkar i *key* och antalet saknade länkar i *response*. Analogt kan samma värde fås om *key* och *response* byter plats, att varje $R_i \in response$ partitioneras med avseende på elementen i *key*. Vi visar nu hur ett element *K* i *key* partitioneras med avseende på elementen i *response*:

Vi låter $I(K_i)$ vara en partition med alla icke-tomma snitt mellan ett element K_i i *key* och elementen i *response*:

$$I(K_i) = \left[x \mid x = K_i \cap R_j \text{ där } R_j \in response\right].$$

$I(K_i)$ innehåller nu mängder som inkluderar alla element som finns representerade i både K_i och i något R_j. Partitionen skapas nu av unionen av $I(K_i)$ och alla delmängder i K_i bestående av ett element som inte finns i något element i $I(K_i)$. Detta kan uttryckas:

$$p(K_i) = I(K_i) \cup \left(\left[x \mid x \subseteq K_i \text{ och } |x| = 1\right] \setminus \left[x \mid x \subseteq I_{ij} \text{ där } I_{ij} \in I(K_i)\right] \right)$$

Detta innebär att varje element i $p(K_i)$ antingen innehåller ett element (för saknade länkar eller länkar som är utspridda mellan olika element i *response*) eller fler än ett för element där länkarna är identiska i K_i och något R_j. Är K_i identiskt med ett element i *response* kommer $p(K_i)$ att bestå av ett element identiskt med K_i. Vi illustrerar partitioneringen med ett exempel:

$$key = [[1,2,3,4],[5,6],[7,8,9],[A,B,C]]$$
$$response = [[1,2,3,4],[7,8],[9,B],[C,D],[E,F,G,H]]$$

Vi börjar med att partitionera *key* med avseende på *response*. För varje K_i skapas en ny partition:

$$p(K_1) = [[1,2,3,4]]$$
$$p(K_2) = [[5],[6]]$$
$$p(K_3) = [[7,8],[9]]$$
$$p(K_4) = [[A],[B],[C]]$$

K_1 har en direkt motsvarighet i *response* och partitionen för K_1 innehåller därför ett element som är identisk med K_1. Båda elementen i K_2 saknas i alla R_i och därför innehåller partitionen för K_2 två mängder med ett element. Elementen i K_3 finns utspridda i två mängder i *response* och partitionen för K_3 blir därför uppdelad i två mängder. Av elementen i K_4 finns två representerade i olika R_i och ett finns inte representerat så varje element i K_4 bildar en mängd i partitionen. Nu gör vi motsvarande operation för *response* med avseende på *key*:

$p(R_1) = [[1,2,3,4]]$
$p(R_2) = [[7,8]]$
$p(R_3) = [[9],[B]]$
$p(R_4) = [[C],[D]]$
$p(R_5) = [[E],[F],[G],[H]]$

Nu kan vi räkna ut precision och täckning. Vi betecknar med $c(K_i)$ det minimala antalet korrekta länkar som är nödvändigt för att generera ekvivalensklassen K_i, d.v.s. antalet element i K_i minus ett.

$$c(K_i) = |K_i| - 1$$

Med $s(K_i)$ betecknar vi antalet saknade länkar i *response* relativt K_i.

$$s(K_i) = |p(K_i)| - 1$$

För varje ekvivalensklass K_i i *key* blir felet för täckningen $\frac{s(K_i)}{c(K_i)}$, d.v.s. antalet saknade länkar dividerat med antalet korrekta länkar. Täckningen i sin tur blir:

$$\frac{c(K_i) - s(K_i)}{c(K_i)}$$

som kan uttryckas

$$\frac{(|K_i| - 1) - (|p(K_i)| - 1)}{|K_i| - 1} = \frac{|K_i| - |p(K_i)|}{|K_i| - 1}$$

Täckningen totalt blir nu en summering av täckningen för varje ekvivalensklass K:

$$\text{Täckning} = \frac{\sum_{K_i \in key} (|K_i| - |p(K_i)|)}{\sum_{K_i \in key} (|K_i| - 1)}$$

Precisionen beräknas genom att *key* och *response* byter roller i uttrycket:

$$\text{Precision} = \frac{\sum_{R_i \in response} (|R_i| - |p(R_i)|)}{\sum_{R_i \in response} (|R_i| - 1)}$$

Nu kan vi beräkna täckning och precision för exemplet:

$$\text{Täckning} = \frac{\sum\left(|K_i| - |p(K_i)|\right)}{\sum\left(|K_i| - 1\right)} = \frac{(4-1)+(2-2)+(3-2)+(3-3)}{(4-1)+(2-1)+(3-1)+(3-1)} = \frac{4}{8} = 50{,}0\%$$

$$\text{Precision} = \frac{\sum\left(|R_i| - |p(R_i)|\right)}{\sum\left(|R_i| - 1\right)} = \frac{(4-1)+(2-1)+(2-2)+(2-2)+(4-4)}{(4-1)+(2-1)+(2-1)+(2-1)+(4-1)} = \frac{4}{9} \approx 44{,}4\%$$

Att ha två olika mått försvårar jämförelser mellan olika utvärderingar. Därför anger man ofta resultatet av en utvärdering med ett mått, *F-värdet*. F-värdet är det harmoniska medelvärdet mellan täckning och precision. Vi beräknar F-värdet för exemplet:

$$\text{F - värde} = \frac{2 \cdot \text{täckning} \cdot \text{precision}}{\text{täckning} + \text{precision}} = \frac{2 \cdot 1/2 \cdot 4/9}{1/2 + 4/9} = \frac{8}{17} \approx 47{,}1\%$$

A.2 Utvärdering för referensbestämning

Vid integrationen i Carsim identifierades de nominalfraser som refererade till någon referent vars typ fanns definierad i ontologin. Både koreferenskedjor och enskilda nominalfraser identifierades. Metoden i A.1 utnyttjar länkarna i koreferenskedjorna. Detta innebär att metoden inte kan användas för att utvärdera förekomsten av enskilda refererande textelement. För detta krävs en utvärderingsmetod som istället ser på förekomsten/avsaknaden av enskilda element. Baldwin et al. (1997) beskriver en utvärderingsmetod som räknar ut täckning och precision för varje enskilt textelement vilka sedan kombineras för att räkna ut total täckning och precision. På samma sätt som i A.1 låter vi $key = [K_1, K_2, \ldots, K_n]$ beteckna de korrekta koreferenskedjorna och $response = [R_1, R_2, \ldots, R_m]$ beteckna de genererade kedjorna. Med en kedja menar vi här även enskilda refererande element. Vi definierar precision och täckning för varje funnet element, *i*:

$$\text{täckning}_i = \frac{\text{antalet korrekta element i den } \textit{genererade kedjan} \text{ som innehåller } i}{\text{antalet element i } \textit{sanningskedjan} \text{ som innehåller } i}$$

$$\text{precision}_i = \frac{\text{antalet korrekta element i den } \textit{genererade kedjan} \text{ som innehåller } i}{\text{antalet element i den } \textit{genererade kedjan} \text{ som innehåller } i}$$

Figur A.2 Definition av täckning och precision.

Täljaren i uttrycken avser antalet korrekta element med avseende på *i*. Värdet för täckning och precision för hela dokumentet blir nu en summering av elementens värde multiplicerat med en vikt. Man kan låta vikten variera för olika typer av element om man vill ge dem olika prioritet. Det har dock inte gjorts här, alla vikter är lika. Vikterna, *wt* och *wp*, sätts till 1/antal refererande uttryck i *key* respektive 1/antal refererande uttryck i *response*.

$$\text{täckning} = \sum_{i=1}^{\text{N}} wp \cdot \text{täckning}_i$$

$$\text{precision} = \sum_{i=1}^{\text{N}} wt \cdot \text{precision}_i$$

Där $\text{N} = \sum_{R_i \in response} |R_i|$ d.v.s. antal refererande uttryck i *response* och $wt = \frac{1}{\sum_{K_i \in key} |K_i|}$ och $wp = \frac{1}{N}$.

Vi låter $p(K_i)$ vara en partition med alla snitt mellan ett element K_i i *key* och elementen i *response*:

$$p(K_i) = [P_{i1}, P_{i2}, \ldots, P_{in}] = [x \mid x = K_i \cap R_j \text{ där } R_j \in response]$$

Kardinaliteten för varje $P_{ij} \in p(K_i)$ är nu lika med antalet korrekta element i den genererade kedjan. Täckningen för ett enskilt element i P_{ij} kan skrivas (jämför figur A.2):

$$\text{täckning}_{e \in P_{ij}} = \frac{|P_{ij}|}{|K_i|}$$

Det är klart att alla element i ett visst P_{ij} har samma täckning. Nu kan vi räkna ut den totala täckningen.

$$\text{täckning} = wt \cdot \sum_{K_i \in key} \frac{\sum_{P_{ij} \in p(K_i)} |P_{ij}|^2}{|K_i|}$$

Kvadraten på beloppet kommer sig av alla element i något P_{ij} har identisk täckning. För att räkna ut precisionen använder vi istället partitioner på elementen i *response* med avseende på *key*:

$$\text{precision} = wp \cdot \sum_{R_i \in response} \frac{\sum_{P_{ij} \in p(R_i)} |P_{ij}|^2}{|R_i|}$$

Ett exempel:

Key: `(1, 2, 3, 4), (5, 6), (7), (8), (9, A, B)`
Response: `(1, 2, 3)(4, 5) (8) (A, B, C), (D, E), (F)`

Partitionerna blir:

$p(K_1) = [[1,2,3],[4]]$
$p(K_2) = [[5]]$
$p(K_3) = [\,]$
$p(K_4) = [[8]]$
$p(K_5) = [[A, B]]$

$p(R_1) = [[1,2,3]]$
$p(R_2) = [[4],[5]]$
$p(R_3) = [[8]]$
$p(R_4) = [[A, B]]$
$p(R_5) = [\,]$
$p(R_6) = [\,]$

$$\text{täckning} = wt \cdot \sum_{K_i \in key} \frac{\sum_{P_{ij} \in p(K_i)} |P_{ij}|^2}{|K_i|} =$$

$$= \frac{1}{4+2+1+1+3} \cdot \left(\frac{3^2+1^2}{4} + \frac{1^2}{2} + \frac{0}{1} + \frac{1^2}{1} + \frac{2^2}{3} \right) = \frac{1}{11} \cdot \frac{16}{3} = \frac{16}{33} \approx 48{,}48\%$$

$$\text{precision} = wp \cdot \sum_{R_i \in response} \frac{\sum_{P_{ij} \in p(R_i)} |P_{ij}|^2}{|R_i|} =$$

$$= \frac{1}{3+2+1+3+2+1} \cdot \left(\frac{3^2}{3} + \frac{1^2+1^2}{2} + \frac{1^2}{1} + \frac{2^2}{3} + \frac{0}{2} + \frac{0}{1} \right) = \frac{1}{12} \cdot \frac{19}{3} = \frac{19}{36} \approx 52{,}78\%$$

B Ordlista

Här presenteras ett antal termer som använts i arbetet.

Anafor (*anaphor*) – För två korefererande nominalfraser kallas den som kommer sist i texten för anafor.

Antecedent *(antecedent*) – För två korefererande nominalfraser kallas den som kommer först i texten för antecedent.

Beslutsträd (*decision tree*) – Ett beslutsträd är en trädstruktur där varje intern nod svarar mot ett attribut vars möjliga värden vart och ett leder till en ny nod och lövnodernas värde representerar en klass. I det här arbetet har ett beslutsträd där egenskaperna i egenskapsvektorn är attribut och varje exempel, antecedent-anaforpar, som presenteras för trädet klassificeras som antingen korefererande eller icke-korefererande.

Egenskapsvektor (*feature vector*) – För varje par av nominalfraser finns en associerad egenskapsvektor med utvunna egenskaper för paret.

Egenskapsöverföring (*feature transfer*) – En metod att under klustringen överföra det mest specifika egenskapsvärdet för vissa egenskaper till samtliga uppmärkningsbara element som finns i koreferenskedjan.

F-värde (*F-measure*) – F-värdet är det harmoniska medelvärdet mellan precision och täckning: $\text{F-värde} = \frac{2 \cdot \text{täckning} \cdot \text{precision}}{\text{täckning} + \text{precision}}$. Måttet används då det är lättare att tolka en jämförelse för ett mått än för två.

Identitetssrelation (*identity relation*) – Identitetsrelationen är ekvivalensrelation definierad för två korefererande nominalfraser. Relationens egenskaper möjliggör skapandet av koreferenskedjor utifrån mängder av par av korefererande nominalfraser.

Klassificerare (*classifier*) – Någonting som klassificerar objekt i någon av de möjliga klasserna. Exempel på en klassificerare är ett beslutsträd.

Klustringsalgoritm (*clustering algorithm*) – Syftet med en klustringsalgoritm är att partitionera en mängd objekt i kluster med hjälp av något mått på likhet. I det här arbetet har en enkellänkad klustringsalgoritm med höger-till-vänster-sökning använts. Dess uppgift är skapa koreferenskedjor av tillgängliga nominalfraser med hjälp av en klassificerare och en sökningsalgoritm.

Koreferens (*coreference*) – Koreferens för nominalfraser innebär att de har en gemensam referent, att de representerar samma objekt i världen.

Koreferensbestämning (*coreference resolution*) – Koreferensbestämning för nominalfraser är en process för att avgöra om två eller flera nominalfraser hänför sig till samma referent.

Koreferenskedja (*coreference chain*) – En koreferenskedja är en mängd med alla nominalfraser som har en gemensam referent. Koreferenskedjorna för en given text bildar en partition på alla korefererande nominalfraser.

Korpus (*corpus*) – En samling texter på naturligt språk som används i ett språkvetenskapligt sammanhang.

Namngiven entitet (*named entity*) – En namngiven entitet avser egennamn av olika typer i en text. I detta arbete associeras en märkningsetikett till varje funnen namngiven entitet med information om dess typ, t.ex. person.

Nominalfras (*noun phrase, noun group {partial}*) – En nominalfras är en fras i en text som fungerar på samma sätt som ett substantiv i syntaktiskt avseende. Alla nominalfraser har ett huvudord och kan även ha bestämningar före och efter huvudordet, vilka ger ytterligare information om huvudordet.

Ordklasstagg (*part-of-speech tag*) – En ordklasstaggare märker upp ord i en text med ordklasstaggar. Varje ordklasstagg innehåller olika typer av grammatisk information om ordet, t.ex. ordklass, numerus och genus.

Precision (*precision*) – Precision betecknar hur stor andel av de funna entiteterna som är relevanta. För koreferensbestämning innebär detta:

$$\text{precision} = \frac{\text{antalet korrekta länkar i de maskinellt genererade koreferenskedjorna}}{\text{antalet länkar i de maskinellt genererade koreferenskedjorna}}.$$

Repeated hold-out – En metod för utvärdering. Mängden med tillgängliga textdokument delas slumpvis in i två disjunkta mängder där den ena mängden, träningsmängden, används för att generera en klassificerare och utvärderingen sker på den andra mängden, testmängden. Proceduren upprepas under ett antal iterationer med nya slumpvisa urval tills ett stoppkriterium har mötts.

Täckning (*recall*) – Täckning betecknar hur stor andel av de relevanta entiteterna som funnits. För koreferensbestämning innebär detta:

$$\text{precision} = \frac{\text{antalet korrekta länkar i de maskinellt genererade koreferenskedjorna}}{\text{antalet länkar som skulle märkts upp}}.$$

Uppmärkningsbart element (*markable*) – Begreppet avser alla nominalfraser i en text som är kandidater att ingå i en koreferenskedja. Förutom nominalfrasen är även en mängd egenskaper, elementattribut, associerade till det uppmärkningsbara elementet.