Kan IT-system förstå ostrukturerad information?

Arne Jönsson, Linköpings universitet och Santa Anna IT Research Institute

Inledning

Mängden information lagrad i datorer ökar ständigt och det blir alltmer viktigt att dessa informationskällor, såväl lokala som globala, blir enkelt tillgängliga för att stödja våra dagliga aktiviteter. Emellertid är huvuddelen av informationen producerad och lagrad i form av text eller tal, utan strukturer som gör det möjligt för datorer att förstå innehållet. Vidare är de sökmotorer som finns begränsade till att söka på vissa kombinationer av nyckelord och möjligheterna att enkelt formulera komplexa sökfrågor är begränsat.

Språkteknologi är en väsentlig basteknologi för såväl interaktion som för att analysera och strukturera dokument av olika slag. Språkliga analyser gör att texter kan berikas med information som gör det möjligt att förstå vad en text handlar om. Med en dialog på svenska blir det också lättare och mer intuitivt att formulera sitt informationsbehov.

För att illustrera möjligheterna med språkteknologi presenterar vi här några olika projekt som utvecklats vid laboratoriet för databehandling av naturligt språk vid Linköpings universitet i samarbete med bl.a. Ida Infront, Nokia Home Communications och SVT. Vår forskning syftar till att utveckla verktyg för att göra språkliga analyser av texter samt att utveckla dialogsystem där användaren i en dialog med systemet enkelt och intuitivt kan precisera sitt informationsbehov.

FrågaRSV

Det första projektet, FrågaRSV, undersökte möjligheten att utnyttja en enkel ontologi, dvs. en representation av de typer av objekt, deras egenskaper och relationer som finns i en given domän, för att utöka publika elektroniska dokument med domänspecifik information som i sin tur möjliggör enkel frågesvarsinteraktion. Frågesvarssystemet utvecklades för att svara på frågor ur RSVs broschyr ”Dags att deklarera”. Vi utnyttjade här flera språkteknologiska standardtekniker från dokumentkonverteringar till morfosyntaktisk analys. Dokumenten XML-kodades och ett domänspecifikt lexikon skapades. Vi utnyttjade ett kommersiellt tillgängligt analysprogram från Connexor som skapar en struktur över funktionella språkliga beroenden med hjälp av en sk. FDG-parser. Vi skapade också en frågekorpus huvudsakligen från RSVs fråge-svarssida samt en manuell analys av dokumenten.

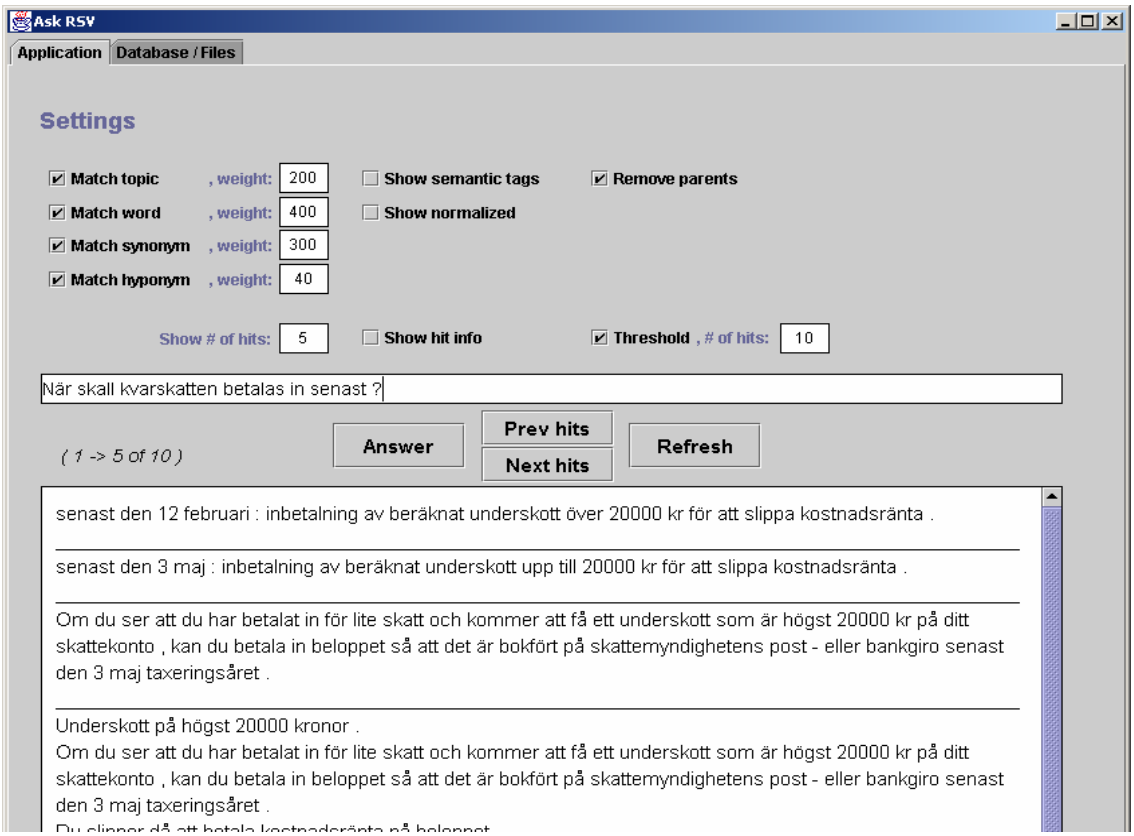
Frågekorpusen bestod huvudsakligen av faktafrågor som inte krävde ytterligare följdfrågor eftersom svaren oftast kunde hämtas som textfragment ur dokumenten. Frågekorpusen visade emellertid att det var vanligt att man använde såväl synonymer som hyponymer som inte fanns i dokumenten. Många begrepp i deklarationsdomänen används olika beroende på om man är expert eller lekman. Begreppet *restskatt*, som ofta används av lekmän, finns till exempel inte längre, det heter *underskott på skattekontot*.

Det lexikon som användes i FrågaRSV skapades automatiskt ur broschyren samt frågekorpusen. Dessutom lade vi manuellt in synonymer och hyponymer till lexikon och gav dem en adekvat beskrivning. Lexikonet utökades också med flerordsuttryck som extraherades med ett program för frasigenkänning. XML-dokumenten märktes sedan upp syntaktiskt med information från FDG-parsern och det domänspecifika lexikonet.

Utöver den syntaktiska informationen märktes dokumentet upp med semantiska markörer. För detta skapades en enkel domänspecifik ontologi med begrepp från skattedomänen. Vi definierade ungefär 100 begrepp som lades till ontologin och som innehöll hyponym- och synonym-relationer. Lexikonet uppdaterades med referenser till de ontologiska objekten. Vi

skapade även ett antal mallar som kunde hantera temporala och numeriska uttryck. Dessa användes sedan tillsammans med ontologin för att märka upp XML-dokumenten med ytterligare semantisk information.

Vid frågetolkningen användes samma ontologi, men istället för att göra en fullständig analys av frågan arbetade vi med sk. bag-of-words-analys. Detta innebär att innehållsorden i frågan mappas mot ontologiska objekt som i sin tur används för att finna lämpliga meningar och avsnitt i dokumenten. Dessa rangordnas beroende på bl.a. ontologiskt avstånd mellan begrepp i frågan och den semantiska informationen i mängden svar som hittats. I figuren nedan visas ett exempel på en fråga till FrågaRSV.



Som framgår av skärmdumpen skriver vi ut oredigerade textavsnitt där ett svar hittats som passar in på de begrepp som matchat.

Vi har inte gjort någon systematisk utvärdering av FrågaRSV. Syftet var främst att se vad man kan åstadkomma med språkteknologiska standardtekniker, en väldigt enkel, och begränsad, ontologi i kombination med en begränsad frågeanalys. FrågaRSV fungerar emellertid väl och ger ofta imponerande svar när vi testar den på vår initiala frågekorpus. Systemet klarar av att hantera frågor om räntor, summor, datum and tidsintervall samt ge svar på frågor om vad begrepp betyder. Däremot klarar FrågaRSV inte av dialog, så man kan inte ställa följdfrågor utan varje fråga måste vara fullständig.

TvGuide

Om informationen är tillgänglig i strukturerad form blir det lättare att söka och vi kan också ägna mer tid åt att förstå den pågående interaktionen. TvGuide var ett sådant projekt där syftet var att skapa ett dialogsystem för att göra det lättare att hitta rätt i framtidens digitala tv-utbud. Med hjälp av TvGuide kan man söka efter tv-program och få fram mer information om t.ex. en långfilm.

Det som främst skiljer TvGuide från FrågaRSV är att TvGuide har förmåga att hantera sammanhållen dialog. Detta innebär att TvGuide håller reda på vad man pratar om så att man

kan ställa följdfrågor utan att behöva upprepa vad man tidigare pratat om. Vidare kan TvGuide begära förtydliganden av användaren. Om man t.ex. frågar vilka filmer som går på tv i kväll blir svaret troligen alldeles för långt, TvGuide begär därför ytterligare information från användaren för att kunna ge ett lagom stort svar. Eftersom TvGuide var kopplad till en Internetbaserad filmtjänst kunde användaren här t.ex. svara med en genre eller skådespelare. Naturligtvis kunde man också fråga efter filmer på vissa kanaler. När svaret sen dykt upp kan man ställa följdfrågor. Följande dialog visar på en interaktion med TvGuide:

```
A: Vilka filmer visas på tv ikväll
S: Det finns många filmer efter klockan 18 ikväll.
   Var vänlig precisera dig.
A: Bara de på TV1 och TV2
S: <Visar en lista av filmer>
A: Vem har huvudrollen i bondfilmen
S: Sean Connery
A: Vem regisserade den
S: Guy Hamilton
A: Ok, vilka nyheter finns
S: <Visar nyheter efter klockan 18>
.
.
```

TvGuide arbetade i själva verket med talad interaktion på engelska, men dialogen ovan är översatt från en autentisk dialog. Som framgår av dialogen tolkar TvGuide temporala uttryck som ”i kväll” till efter klockan 18 samma dag. TvGuide tolkar också yttranden baserat på den tidigare interaktionen och minns användarens preferenser, som t.ex. att nyheterna i sista yttrandet troligen skall vara ikväll efter klockan 18.

BirdQuest

Den stora utmaningen är att koppla samman dessa båda språkteknologiska forskningsområden så att man kan skapa system som tillåter dialog från databaser som skapats automatiskt ur vanliga dokumenttexter. För att lyckas måste de kunskapskällor som används, främst lexikon och ontologi, måste vara lika så att systemet kan förstå vad användaren menar med sina frågor i termer av den information som extraherats ur dokumenten.

BirdQuest är ett sådant system som skapats för att man skall kunna ställa frågor på internet i anslutning till naturprogram på TV. Vi utgick här ifrån fågelboken Nordens Fåglar samt en korpus av 329 frågor som samlats in via SVT:s webbsidor.

Informationsextraktion

För att extrahera informationen ur fågelboken märktes den upp med ontologisk information på i princip samma sätt som för FrågaRSV. Vi skapade enkla mönster för att identifiera namnenheter såsom fågelnamn, färger, mått etc. Ett lexikon skapades för domänen med hjälp av ett verktyg för informationsextraktion. En första enkel ontologi utvecklades med representationer för objekttyper, deras egenskaper och relationer och fågelboken märktes upp med XML-taggar.

Det visade sig ganska tidigt att de frågor man ställde till BirdQuest ofta innebar mer komplexa sökningar än de som ställdes till FrågaRSV, t.ex. frågan *Vilken fågel är störst?* leder till att vi måste skriva en funktion som kan söka igenom hela fågelboken och plocka ut den största. Nästa naturliga steg var därför att undersöka möjligheten att konvertera XMLdokumenten till en databas för att därigenom kunna utnyttja de inferensmöjligheter som SQL tillhandahåller.

Vid transformationen till en databas valde vi att extrahera information ur dokumenten med större precision, baserat på frågekorpusen. Ett stort antal mönsterigenkänningsregler användes för att identifiera relevant information som kunde överföras till relationsdatabasens attribut-värde-format. Syftet var att bara fylla databasen med relevant information och att utesluta textfragment som inte efterfrågades i frågekorpusen.

Som ett exempel på detta ger följande utdrag ur fågelboken

```
Smålom (Gavia stellata)
Utseende: 53-69 cm, vingbredd 106-116 cm. Obetydligt mindre än storlommen
   men slankare byggd. Sommartid utmärks smålommen av sin jämnbruna rygg
   och brunröda strupe. Hals och näbb markant smalare än storlommens.
   Näbben hålls ofta något uppåtriktad. Vintertid är smålommen ljusare och
   ej så kontrastrik som storlommen. Ryggen är beströdd med en mängd små
   vita prickar. Ungfågeln liknar den gamla i vinterdräkt men har mindre
   fläckad rygg och gråare huvud och hals ...
```

den extraherade information nedan:

```
NAME: smålom
LATIN_NAME: Gavia stellata
MAX_WING: 116
MIN_WING: 106
MAX_HEIGHT: 69
MIN_HEIGHT: 53
WINTER_SHAPE: Ungfågeln liknar den gamla i vinterdräkt men har mindre
   fläckad rygg och gråare huvud och hals ...
```

Extraktionen har alltså premierat precision för att få färre felaktiga svar, vilket betyder att extraktionen ibland tar bort ganska mycket information. Till exempel utelämnas ofta information om egenskaper där andra fåglar används som referens, som i exemplet ovan. Databasen beror mycket på hur bra textextraktionskomponenten är. Ju mer avancerad denna är desto mer information kommer att finnas i databasen. I nuvarande version av BirdQuest har all information extraherats automatiskt utifrån de mönster som skapats, vilket ofta ger upphov till hål i databasen.

Databasextraktionen gjorde att BirdQuest i princip fick samma inferensmöjligheter som ett traditionellt dialogsystem, under förutsättning att den rutin för att transformera frågor till SQL kunde extrahera tillräckligt mycket information ur den kontextuellt tolkade frågan från användaren.

Dialogkomponenten

Dialogkomponenten i BirdQuest kontrollerar interaktionen och hanterar dialoghistorien. Den är utvecklad från MALIN, ett ramverk för att skapa dialogsystem utvecklat vid vårt forskningslaboratorium. Speciellt för MALIN är att det innehåller, utöver tolkare för naturligt språk och dialoghanterare, en modul för domänkunskapshantering som underlättar access till olika bakgrundssystem och byte till olika domäner.

Dialogen representeras i MALIN i dialogobjekt med parametrar för att representera de objekt, i fallet BirdQuest är objekten oftast fåglar, och de egenskaper, som t.ex. storlek, som är relaterade till ett objekt. Det är inte nödvändigt att specificera båda parametrarna. BirdQuest utnyttjar dialoghistorien för att kontextuellt tolka yttranden som är underspecificerade.

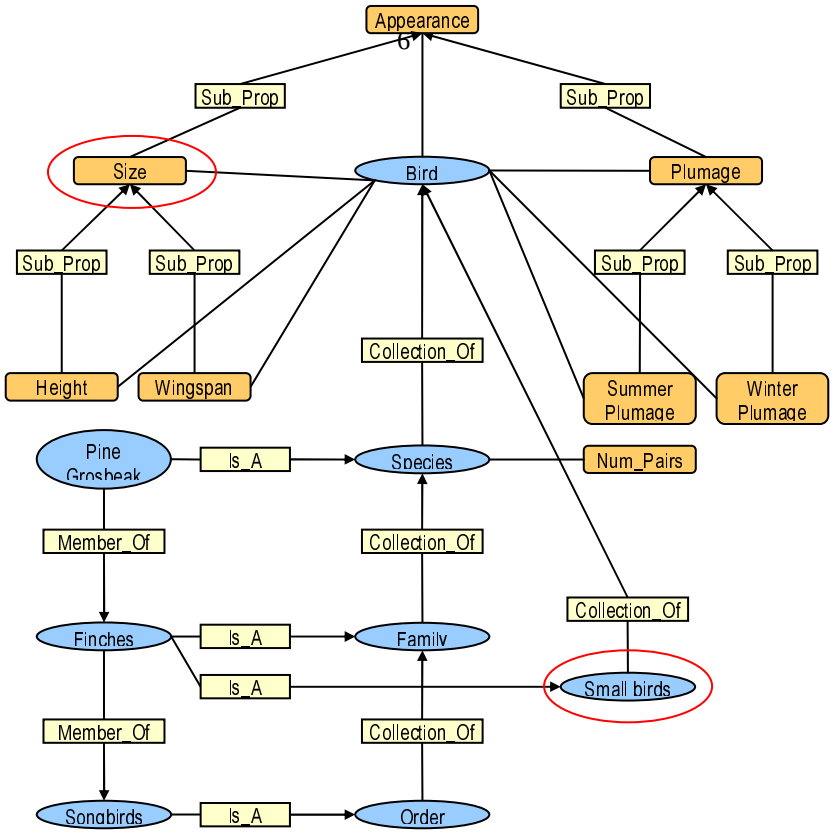
I figuren illustreras en del av de dialogfenomen som BirdQuest hanterar.

I det första yttrandet, *Vilka fåglar häckar i afrika,* utnyttjar BirdQuest de inferensmöjligheter som finns i SQL för att besvara en komplex sökfråga. Användaren följer sedan upp detta med att fråga om en egenskap hos en av de fåglar som finns i svaret. I fokus finns nu det objekt och den egenskap som efterfrågades i förra frågan, dvs. egenskapen ”storlek” för objektet ”tornfalk”. Detta används för att tolka frågan *hur många ägg lägger den* som hur många ägg tornfalken lägger. Nästa inlägg från användaren är ett elliptiskt yttrande, *Lom,* som med hjälp av dialoghistorien tolkas som *Hur många ägg lägger en lom*. Nu finns det flera olika arter av släktet lom, vilket skulle ge ett för stort svar att presentera. Därför begär dialoghanteraren ytterligare information genom att tala om vilka arter som finns och be användaren precisera sin fråga. Efter svaret på den frågan är smålom objekt i fokus och antal ägg egenskapen i fokus och dialogen fortsätter.

Ontologikonstruktion

Som nämnts tidigare har användare ofta en annan syn på vad ord och begrepp betyder jämfört med hur de används i det referensmaterial som skrivits av domänexpeterna, i fallet BirdQuest ornitologernas syn på fåglar och dess egeneskaper. I fågelboken finns t.ex. ett antal olika typer av fjädrar som aldrig används av de som inte är väldigt insatta. BirdQuest måste därför kunna hantera såväl användarens syn på världen, dvs användarontologin, som en domänontologi baserad på de textuella dokumenten. Den förra används av dialoghanteraren för att tolka användarnas yttranden medan den senare används av domänkunskapshanteraren för att skapa SQL-frågorna till relationsdatabasen.

Dessa båda ontologier måste sedan kunna kombineras och konflikter hanteras. Detta görs på lite olika sätt. I figuren visas hur ontologin utökats med två användarbegrepp, inringade, och hur dessa länkats in.



Begreppet ”stor” används ofta av användarna, t.ex. i frågor som *Vilken är den största fågeln?*. I fågelboken finns emellertid inte begreppet stor, utan man pratar där om vingbredd och längd. I ontologin införs därför ett överordnat begrepp ”Size” som i sin tur har begreppen ”Height” och ”Wingspan” som underbegrepp. Detta är naturligtvis domänberoende eftersom t.ex. att vara stor inte innebär att ha stor vingbredd i alla domäner. Ett annat exempel är begreppet småfåglar som också saknas i domänontologin men som finns i användarontologin. Här kan vi inte skapa ett överordnat begrepp utan inför istället ett nytt sidobegrepp i ontologin ”Small birds”.

Utvärdering

BirdQuest är utvärderat i en mindre undersökning då 27 användare, utan kunskap om dialogsystem och mestadels litet intresse av fåglar, provade BirdQuest utan andra instruktioner om systemet än att man kunde fråga om nordens fåglar. Sammanlagt skrev de in 518 yttranden, såväl egna frågor som svar på systemets klargörandefrågor. En hel av dessa var utanför domänen, som t.ex. *Hur bli man ornitolog?* men totalt lyckades BirdQuest ge ett riktigt svar på 35% av alla användarnas frågor. Vidare var 13% klargörandefrågor från systemet, vilket visar att dialog är väsentligt. BirdQuest kunde således inte svara på eller tolka 52% av användarnas frågor, ofta beroende på att frågorna var utanför domänen. Totalt 23% av användarnas frågor var sådana.

Många av de frågor som BirdQuest trots allt inte kunde besvara berodde på att ontologin inte var fullständig vilket resulterade i att BirdQuest inte kunde tolka en fråga eller att det inte fanns någon information i databasen. Ett vanligt problem gällde färger, där många användare frågade efter färgen på en fågel. Problemet här är att en fågel i fågelboken inte har en färg utan färger är istället beskrivna i termer av de olika kroppsdelarna och målande färgnyanser, t.ex. ”Honan är mörkgrön med ljus strupe, vingframkant och hjässa. Ungfåglarna liknar honan, men är mättat chokladbruna med rostgul hjässa och strupe …”. En möjlighet skulle vara att ge hela textfragmentet, men det är inte alltid som utseendebeskrivningen utnyttjar färger, ibland är det t.ex. storleken eller vingformen som beskrivs som utmärkande.

Sammanfattning

Vi har genom några olika system visat att det är möjligt att skapa dialogsystem som kan besvara frågor om information som från början finns i vanliga textdokument. De presenterade systemen är prototyper och långtifrån kompletta. En av de stora utmaningarna ligger i att skapa

den ontologi som beskriver domänen. Här har vi i våra forskningsprototyper arbetat på egen hand. Med en domänexpert blir naturligtvis ontologin korrektare och kan bättre stödja informationsextraktionen.

Extraktionsarbetet sker iterativt, dvs. vi skapar ett mönster för att se vilken information det extraherar och när vi är nöjda med mönstret kör vi det på några fler dokument, förfinar ytterligare innan den slutliga extraktionen sker. För närvarande arbetar vi med att utveckla ett verktyg som skall underlätta detta arbete så att det skall bli enklare att byta domän. Vidare arbetar vi med att utveckla verktyg för att skapa ontologier som kan spegla såväl användarens som domänens användning av olika begrepp.

Det kommer då att bli enklare att märka upp dokumentsamlingar av t.ex. myndighetstexter med semantisk information så att sökningen kan bli mer sofistikerat än med dagens sökmotorer. Med ett dialogsystem kopplat till sådana strukturerade texter kommer det också att bli enklare och mer intuitivt att formulera komplexa sökfrågor. Tillgängligheten ökar och vi är en bit på vägen mot t.ex. en 24-timmarsmyndighet där medborgare kan nå relevant samhällsinformation.

Mycket av den programvara som produceras finns tillgänglig som öppen källkod på http://nlpfarm.sourceforge.net/

Tack

Vår forskning finansieras av Vinnovas program för språkteknologi, Centrum för industriell informationsteknologi (Ceniit) samt Svenska IT Institutet (SITI). Tack till Frida Andén, Mikael Andersson, Lars Degerstedt, Annika Flycht-Eriksson, Pontus Johansson, Magnus Merkel, Sara Norberg, Michael Petterstedt och Håkan Sundblad.