Ett gränssnitt med Naturligt språk till en TV programs-databas

Mikael Hallin
Department of Computer Science
Lund University
mikaelhallin@msn.com

Samuel Andersson
Department of Computer Science
Lund University
tetsu@algonet.se

Abstract

This report describes how we implemented a TV guide application that is controlled by natural language. The application is using simple strings for its input and output. The idea around the language processing part is to make it simple to implement. We are using keywords to parse the questions and the answers are then generated by templates.

1 Inledning

Denna rapport beskriver hur man kan applicera ett användargränssnitt med naturligt språk för att söka i en databas med information om TV-program. Projektet delades upp i tre logiska moduler, vilka är helt oberoende av varandra. De tre modulerna är en TV-programs-databas (TVDB), en språkbehandlingsmotor och ett GUI. Datan för TVDBn hämtas från olika webbplatser, beroende på vilket land som TV-programmen visas i. Språkbehandlingsmotorn är skriven i Java och har i sin tur egna små databaser med nyckelord. Slutligen skrevs ett enkelt GUI i Swing. Kommunikationen mellan de olika modulerna sker till stor del med en egendefinierad kommandosträng och programlistningar. Till och från slutanvändaren består kommunikationen av naturligt språk blandat med programlistningar.

2 Metoder

Vi började med att identifiera några nyckelfrågor som vi ville att programmet skulle kunna besvara. Exempel på frågor som vi tyckte var viktiga är "När börjar Simpsons?", "Går Seinfeld idag?" och "Går det några filmer ikväll?". Dessa frågor är ganska enkla, men det skulle visa sig senare att den strategin vi valt är väldigt modulär och klarar även mer komplicerade frågor som "Går det någon fotboll den här veckan på TV3?".

För att enkelt få tag på information om TV-program använder vi oss att ett befintligt projekt som heter XMLTV. XMLTV tillhandahåller diverse verktyg som är nyttiga för att hantera TV-tablåer. Projektet tillhandahåller även en hel del skript som kan ladda ner TV-tablåer via internet på ett tio-tal olika språk.

Eftersom programmet bygger på att man ska efterlikna en konversation med datorn, ville vi få GUIt att påminna om vanliga populära instant messaging klienter som t.ex. MSN Messenger och ICQ. Programmets huvudsakliga komponent är därför en stor textyta där konversationen hamnar, med växlande färger för användarens och datorns fraser. Nedanför denna yta finns ett fält där användaren kan skriva in sin fråga och en knapp för att "skicka", d.v.s. starta behandlingen av det användaren har sagt.

3 Implementation

Vi bestämde oss för att implementera projektet i Java, eftersom det går snabbt att jobba

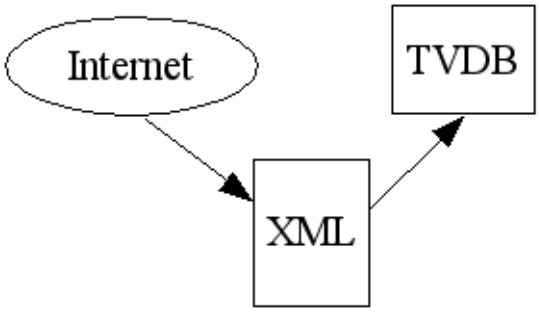

Figure 1: Datans väg från internet till vårt program.

med och har ett omfattande klassbibliotek. De delar vi hade störst nytta av var paketen rörande reguljära uttryck, XML parsning och Swing för att bygga GUIt.

Får att få tag på data som vi kunde använda för att testa programmet använde vi XMLTVs nerladdnings-skript. Vi använder oss av deras svenska skript som hämtar sin data ifrån DagensTV.

3.1 Databasen

Eftersom vi använder den mellanlagrade XML-datan får vi den stora fördelen att vårt program bara behöver förstå ett TV-tablå format. Detta innebär i sin tur att vi bara behöver skriva en parser som läser in datan. För att läsa in datan i minnet används Javas inbyggda SAX parser.

Datan lagras i en XML-fil, enligt en speciell mall, en s.k. Document Type Definition (DTD), som är specificerad av XMLTV-projektet. Mallen är ganska utförlig och kan förutom titel och beskrivning även innehålla information om vem som har regiserat och vilka skådespelare som är med i filmen.

3.1.1 Sökning

Databasen vi arbetar med innehåller programlistningar från cirka 15 kanaler en vecka fram i tiden. Detta motsvarar ungefär 3.000-4.000 poster i databasen, vilket får anses som en relativt liten databas. Vi tyckter därför inte att det är nödvändigt att implementera några avancerade algoritmer för indexering och sökning. Programmet som implementerades söker därför sekvensiellt och går igenom samtliga poster vid varje sökning.

Ett "TV-dygn" är inte som ett vanligt dygn som slutar klockan 12 på natten. Tekniskt sett går ett program som börjar klockan 0:00 på natten på en annan veckodag, men vi ser det ofta som att programmet går sent på kvällen. Vi blev därför tvungna att manipulera datum och klocka för att klara av sökningar på dagar. Vi omdefinierade därför dygnet att gå mellan 04:00 - 04:00. På så vis hamnade program som börjar efter klockan 12 på natten på "rätt" dag. Förutom sökningar på absoluta klockslag vill man även göra sökningar på mer abstrakta tidpunkter som t.ex. kväll eller eftermiddag. Uttrycket kväll definierades därför till ett intervall mellan 18:00-22:00.

3.1.2 Kategorier

Databasen vi fick tag på innehöll inte någon information om vilka kategorier som programmen tillhörde. Vi blev därför tvungna att själva skriva ett antal enkla regler för kategorisering av programmen. För att göra detta tittade vi i programmets beskrivning. I beskrivningen står det ofta saker som "Amerikanskt action-drama från 1991". Vi gjorde därför regler som flaggade programmen som filmer om den här meningen innehöll ord som komedi, thriller, action osv. Program som innehöll ishockey eller NHL, hamnade i hockey-kategorin.

Man kan sedan söka efter program kategorivis, till exempel all fotboll som sänds. Våra kategorier har även visst stöd för hierarkier. Tanken är att man ska kunna söka efter sport och sedan få upp alla sportrelaterade program. Vi hann dock inte bli klara med detta steg, utan nöjde oss med en platt struktur.

3.2 Språkbehandling

Vår plan med programmet är att det skulle klara av att både förstå och svara med naturligt språk. För att klara det på den korta tid vi hade för projektet, valde vi att göra en så enkel parsning som möjligt för texttolkningen. Vi valde därför en slags ny-

ckelordssökning och bryr oss inte så mycket om grammatiken i meningarna. För utmatning gjorde vi färdiga mallar för svar på de olika typer av frågor som programmet stödjer. Mallarna fylls i med det som hittas i TVDB. Internt i programmet så kommunicerar de olika modulerna med varandra med hjälp av programlistningar och en egendefinierad kommandosträng, kallad Commandstring. Kommandosträngen är uppbyggd av informationen som parsas från frågan.

3.2.1 Nyckelord

För att få ut vad det egentligen frågas efter så hade vi idéen att man skulle leta efter nyckelord i meningen. Nyckelordet kan även vara en kort fras. Nyckelorden delas upp i olika grupper beroende på deras betydelse. De grupper vi har är

- Frågeord. Den här gruppen innehåller ord som oftast brukar finnas med i en fråga.
- Skådespelare. Här finns alla skådespelare, kan även använda smeknamn som nyckel.
- Kanaler. Alla kanalerna som finns med i TVDB.
- Dagar. Allt som har med dagar att göra, både namn och benämningar som "idag".
- Programtyper. Olika typer av program, som film och dokumentär.
- Tider. Klockslag och tidsbenämningar. T.ex "eftermiddag" och "ikväll".
- Titlar. Alla titlar som finns i TVDB.

Varje grupp av nyckelord lagras i en egen fil, där varje nyckelord kopplas till ett standardiserat ord eller ett namn som skall användas internt i programmet. Genom att ha standariserade ord så kommer det vara likadant internt oberoende av vilket språk som används för inmatning. Filstrukturen är gjord så att det skall vara enkelt att stöjda flera olika språk, t.ex. för de svenska tidsorden

```
database/se/times.data
```

och såhär kan de första raderna i filen se ut

```
klockan (\d+:\d+) = #1
(\d+:\d+) = #1
ikväll = evening
i kväll = evening
kväll = evening
```

Det till vänster om likhetstecknet är nyckelorden och det till höger är de standardiserade orden eller namnen. Sjävla nyckelordet kan även vara ett regulärtuttryck. Att vi la till den funktionalliteten var för att vi skulle kunna skapa nyckelord som matchar mot vissa mönster. Mönstermatching kommer till nytta på ord som har en viss betydelse, men som kan se olika ut från fall till fall som t.ex. klockslag. Det till höger kan även referera till en regulärtuttrycksgrupp. Det görs genom att skriva # följt av gruppnummert. På det viset kan man få med saker i standardordet som har matchats i inmatningssträngen.

Efterhand som man letar nyckelord tar man bort alla träffar från inmatningssträngen. Att vi gör det är för att det inte skall bli två träffar på samma ställe. Så ordningen man väljer på nyckelorden har stor betydelse. Först ordningen på nyckelordsgrupperna och sedan kan det även ha betydelse på ordningen av nycklarna inom grupperna. För grupper valde vi att ha de först som representera namn, för att i t.ex. filmtitlar var det lätt hänt att de titlarna kunde innehålla ord som dagar och frågeord. Om man då råkat ta bort en del av en titel skulle man aldrig kunna hitta vilken film frågan gällde.

Efter all parsning har man fått ihop en mängd standardord och namn, som man sätter in i Commandstringen.

3.2.2 Commandstring

För att förenkla för sökmotorn till TVDB och för att få någon typ av standard som inte skulle vara så beroende på språk och formulering av frågor, skapade vi Commandstring. Commandstringen är uppdelad i olika segment, där varje segment i princip representerar en nyckelordsgrupp. Alla segmenten behöver inte vara ifyllda för att TVDB-sökmotorn skall

kunna förstå vad den skall leta efter. Själva Commandstringen ser ut på följande sätt

```
command|time|day|channel|progtype|
title|actors
```

De flesta fält är mer eller mindre självförklarande, så command är den enda som kommer att beskrivas i detalj. Command är det segemnt som talar om vad det är för typ av fråga. Oftast så bestämms command genom en matchning av ett frågeord, men ibland så bestäms det beroende på vilka andra segment som är ifyllda eller inte. De commands som finns är

- asktime. Frågar vilken tid något går.
- askwho. Frågar vilka som är skådespelare eller värdar för en produktion.
- asktitle. Frågar titel på en produktion.
- nonsense. Om inte något vettigt kommando kunde hittas.

Några exempel på hur en Commandstring kan se ut

```
Går det någon komedi klockan 20:30
idag?
asktitle|20:30|today||comedy||

Vilken tid går seinfeld imorgon?
asktime||tomorrow|||seinfeld|
```

3.2.3 Mallar

För att generera svar till användaren på ett enkelt sätt valde vi att skapa ett mallsystem. På det viset skulle det inte krävas någon komplicerad kod för att skapa svar i formen av naturligt språk. Genom att dela upp mallarna i grupper, beroende på vad det är för typ av fråga man skall generera svar till, är det lätt att hämta rätt mall. Även mallarna ligger i filer, där filerna har samma logiska filstruktur som nyckelorden. Inom varje mallgrupp finns det tre huvudsakliga varianter av mallar. De tre varianterna är om man hittar många program, ett program eller inga program. Men

Figure 2: Screenshot av GUIt.

där kan också finnas flera mallar för varje huvudvariant, som ser lite olika ut. Det för att man skall kunna få lite olika formuleringar på samma svar, bara för att få programmet att verka mer levande.

Alla mallarna börjar med en bokstav som talar om vilken huvudvariant som den tillhör. p för många program, s för ett program och n för inget program. Själva mallen är bara en vanlig mening där man stoppat in lite markörer på ställen där man vill att det skall stoppas in frågespecifika ord. T.ex.

```
s #title går #time på #channel
```

Vilket t.ex. kan producera svaret

```
seinfeld går 23:40 på ZTV
```

till användaren.

3.3 Användargränssnitt

Användargränssnittet är enkelt och intuitivt uppbyggt. Figur 2 visar ett screenshot av hur det ser ut när man kör programmet. I menyn kan man ändra språk på både in- och utmatning. Vi implementerade stöd för både svenska och engelska för att visa hur flexibelt systemet är.

4 Resultat

Prototypen vi fick fram var funktionell och användbar. Den klara av att parsa de testfrågor vi skapade i början av projektet och till vår glädje så klarar den även av mer komplicerade frågor utan vi behövde göra något utöver vad vi gjorde för att klara testfrågorna. Programmet klarar av att svara tillfredsställande på de flesta frågorna, men i enstaka fall kan svaren ibland bli lite felaktiga.

Vi hann även implementera så att programmet klarar av flera språk, ändringen för att göra det var inte stor. I princip var det att lägga till en meny och ändra lite i filstrukturen för att få plats för filer på flera språk.

5 Slutsatser

Valet vi gjorde med att ha nyckelord och databaser var lyckat. Det var enkelt att implementera och det är enkelt att lägga till nya nyckelord. Oftast när vi skulle utöka stöd för nya frågor så var det bara att ändra i databasen. Det kunde bli ganska avancerade förändringar bara genom att ändra i databasen, vilket var bekvämt då man slapp kompilera om hela tiden.

Mallarna var också ett lyckat drag, även om vi inte hann testa ut dem ordentligt på grund av tidsbrist, så fick vi de att fungera bra.

Referenser

DagensTV.com Website som har TV-tablåer. http://www.dagenstv.com/

XMLTV Projekt som har en mängd verktyg för att hantera TV-tablåer. http://xmltv.sourceforge.net/

Acknowledgement

Vi vill tacka Pierre Nugues för hans goda ideér under projektets gång.