



UPPSALA
UNIVERSITET

Statistiska språkmodeller med klass

*Experiment med modifiering av träningsdata för
statistiska språkmodeller*

Ida Weidenmark

Institutionen för lingvistik och filologi
Språkteknologiprogrammet
Examensarbete i datorlingvistik, 30 hp

18 januari 2010

Handledare:
Joakim Nivre, Uppsala universitet
Stefan Eriksson, TeliaSonera
Johan Westermarck, TeliaSonera

Sammandrag

Denna uppsats undersöker om gruppering av ord i träningsdata kan höja andelen korrekt kategoriserade yttranden i ett dialogsystem av typen fritt tal. Systemet använder Nuance Recognizer 9.0 för taligenkänning och kategorisering. Nuance Recognizer 9.0 har stöd för gruppering av ord i träningsdata och kallar grupperingen för klasser. Klasserna har tagits fram med grundläggande domänkunskap. Femton modeller med klasser och en baseline-modell utan klasser har utvärderats med avseende på både taligenkänning och kategorisering. Taligenkänningen har utvärderats med måttet word error rate och kategoriseringen har utvärderats med måtten täckning, precision och F-score. Utvärdering av kategoriseringen har gjorts både med resultatet av taligenkänning samt på transkriberade yttranden.

Resultaten ger inte något direkt stöd för slutsatsen att klasser bidrar till en förbättrad modell. Resultaten ger inte heller något direkt stöd för den motsatta slutsatsen. Det är dock möjligt att en studie med mer data samt dessa eller andra klasser skulle kunna påvisa en positiv effekt. Resultatet tyder även på att vissa typer av klasser förbättrar taligenkänningen och andra typer av klasser förbättrar kategoriseringen. Uppsatsen har utförts i samarbete med TeliaSonera i Uppsala.

Abstract

This thesis examines whether grouping of words in the training data can increase the proportion of correctly categorized utterances in a spoken dialogue system that uses free speech. The system in this experiment uses Nuance Recognizer 9.0 for speech recognition and categorization. Nuance Recognizer 9.0 has support for grouping of words in the training data. This kind of grouping is called classes. Classes have been developed using basic domain knowledge. Fifteen models with classes and a baseline model without classes have been evaluated according to the metric word error rate for speech recognition and the metrics recall and precision for categorization. Evaluation of categorization has been made both with results of speech recognition and with transcribed utterances.

The results provide no direct support for the conclusion that classes contribute to a better model. The results provide no direct support for the opposite conclusion. However, it is possible that a study with more data, and these or other classes could detect positive effect. The results also suggest that certain types of classes improve speech recognition and other types of classes improve categorization. The thesis work has been carried out in cooperation with TeliaSonera in Uppsala.

Förord

Jag vill först och främst tacka mina handledare Joakim Nivre vid Institutionen för lingvistik och filologi, Uppsala universitet, Stefan Eriksson, TeliaSonera och Johan Westermarck, TeliaSonera för mycket värdefull handledning, teknisk assistans samt stort engagemang i detta projekt. Tack för stort stöd och för att ni alltid tagit er tid för mina frågor.

Ett stort tack till alla på TeliaSonera i Uppsala, speciellt tack till Erik Näslund för att ha initierat detta projekt, givit förslag på ämne och för korrekturläsning, Mathias Johansson för engagemang vid uppstarten av detta projekt, Fredrik Engberg för att ha fått tagit del av tidigare arbete med Nuance Recognizer 9.0, Alf Bergstrand för hjälp med kategorisering, Stefan Haglund för korrekturläsning och Eva Lovén för stöd och uppmuntran.

Tack även till Per-Erik Malmström för gott samarbete, Per Starbäck för goda råd vad gäller Latex och för korrekturläsning, Hans Frimmel och Michel Rowinski för stöd under slutfasen av projektet. Slutligen vill jag tacka Magnus Brogie för hjälp med språkgranskning och för att ha varit ett enormt stöd under hela projektet.

Innehåll

1	Inledning	8
1.1	Syfte	8
1.2	Disposition	8
2	Bakgrund	10
2.1	Taligenkänning	10
2.1.1	Språkmodellering	11
2.1.2	Klassbaserade n -gram-modeller	13
2.2	Kategorisering	13
2.3	Call routing	14
2.4	Utveckling av ett fritt tal-system	15
2.4.1	Datainsamling	15
2.4.2	Utveckling av kategoristruktur	15
2.4.3	Dialogstruktur	16
2.4.4	Pilot, driftsättning och underhåll	16
2.4.5	Teknisk översikt av TeliaSoneras system	17
2.5	Nuance Recognizer 9.0	18
2.5.1	Statistisk språkmodell	18
2.5.2	Statistisk semantisk modell	18
2.5.3	Wrappergrammatik	19
2.6	Varför klasser?	19
2.6.1	Implementering av klasser	20
3	Metod	21
3.1	Definition av klasser	21
3.2	Data	21
3.2.1	Transkribering	22
3.2.2	Kategorisering	22
3.2.3	Datamängder	22
3.3	Metod för utvärdering	23
3.3.1	Utvärderingsmått	23
3.3.2	Testsystemet	24
4	Klasser	26
4.1	Beskrivning av klasserna	26
4.2	Modeller	27
5	Utvärdering	28
5.1	Utvärdering av kategorisering med perfekt taligenkänning	28

5.1.1	Kategorier som förändras med klasser i kategorisering med perfekt taligenkänning	29
5.1.2	Yttranden med klasser	30
5.2	Utvärdering av kategorisering med taligenkänning	31
5.2.1	Kategorier som förändras med klasser i taligenkänning och kategorisering	33
5.2.2	Yttranden med klasser	34
6	Slutsats	36
6.1	Resultat	36
6.1.1	Kategorisering med perfekt taligenkänning	36
6.1.2	Kategorisering av resultatet av taligenkänning	36
6.1.3	Klasser	37
6.2	Fortsatt arbete	37
	Litteraturförteckning	39

Figurer

2.1	Översikt av ett av TeliaSoneras system	17
2.2	Call routing med Nuance Recognizer 9.0.	18
3.1	Översikt av utvärderingssystemet för taligenkänning	25
5.1	Utvärdering av kategorisering med perfekt taligenkänning	28
5.2	Utvärdering av kategorisering med taligenkänning	31
6.1	Modell som endast implementerar klasser i kategoriseringen	38
6.2	Modell som endast implementerar klasser i taligenkänningen	38

Tabeller

2.1	Exempel på tänkbara klasser i olika typer av domäner.	20
3.1	Antal yttranden i olika datamängder	23
4.1	Antal medlemmar i klasserna samt frekvens i datamängderna. . . .	26
5.1	Resultat för utvärderingen av kategorisering med perfekt taligenkänning	30
5.2	Kategorisering med perfekt taligenkänning av yttranden med klasser	31
5.3	WER för samtliga modeller samt resultat av kategorisering med taligenkänning	33
5.4	Samtliga yttranden som tilldelades klasser.	35

1 Inledning

Call routing är ett specialfall av textkategorisering där texterna består av telefonyttranden. En kategoriserare ska tilldela telefonyttrandena en kategori givet en mängd fördefinierade kategorier. Dessa kategorier motsvarar hur samtalet ska styras. Yttrandena är resultatet av taligenkänning. Fritt tal innebär att yttrandena inte behöver vara på en viss form och accepteras av en viss grammatik. Detta kan ses i motsats till styrt tal eller knapptryckningar med DTMF (dual-tone multi-frequency). Fritt tal har ingen traditionell grammatik utan har istället en statistisk språkmodell. Den statistiska språkmodellen tar fram den mest sannolika textsträng som yttrandet motsvarar. För att konstruera en språkmodell behövs en stor mängd data. Hur mycket data man än lyckas samla in kommer man alltid att ha för lite. Ett sätt att skydda sig mot det faktum att man alltid har för lite data är att gruppera ord i klasser. Orden hanteras av språkmodellen som ett ord. Detta är särskilt fördelaktigt för ord med låg frekvens.

1.1 Syfte

Syftet med uppsatsen är att undersöka hur införandet av klasser i träningsdata påverkar den statistiska språkmodellen för taligenkänning och kategorisering i ett system för fritt tal. Taligenkänningen kommer att utvärderas med måttet word error rate och kategoriseringen med måtten täckning och precision. Utvärdering av kategoriseringen kommer att göras både med resultatet av taligenkänning samt på transkriberade yttranden. Uppsatsen har utförts i samarbete med TeliaSonera i Uppsala. Utvärderingen har skett på ett av TeliaSoneras system. Av sekretesskäl kommer inte exakt domän att anges och varken klasser eller kategorier kommer benämnas vid sina rätta namn.

1.2 Disposition

Efter detta inledande kapitel följer kapitel 2 som ger en bakgrund till uppsatsen. I bakgrundskapitlet ges en kortfattad introduktion till taligenkänning baserat på en statistisk modell och hur gruppering av ord i klasser kan bidra till en bättre statistisk språkmodell. Bakgrundskapitlet ger även en introduktion till textkategorisering, call routing och begreppet fritt tal som dialogtyp. I kapitel 3 beskrivs metoden för experimentet och den data som har använts. I detta kapitel beskrivs även det testsystem som använts för utvärdering av modellerna. Kapitel 4 beskriver de klasser som använts i experimentet. Här beskrivs även de modeller som utvärderats. I kapitel 5 diskuteras och redovisas resultatet av

utvärderingarna. Uppsatsen avslutas med kapitel 6 som innehåller en sammanfattning av slutsatserna från kapitel 5, samt förslag till vidare forskning inom ämnet.

2 Bakgrund

I detta kapitel ges bakgrund till experimenten. Kapitlet inleds med en bakgrund till taligenkänning baserat på en statistisk modell i avsnitt 2.1. Avsnitt 2.2 innehåller en bakgrund till kategorisering. Avsnitt 2.3 innehåller bakgrund till call routing som är en sammansättning av taligenkänning och kategorisering. I avsnittet beskrivs även begreppet fritt tal. Avsnitt 2.4 beskriver utveckling av ett fritt tal-system. I detta avsnitt ges även en översiktlig bild av hur ett av TeliaSoneras system är uppbyggt. Därefter beskrivs i avsnitt 2.5 Nuance Recognizer 9.0 som används i experimenten för taligenkänning och kategorisering. Bakgrundskapitlet avslutas med avsnitt 2.6 om klasser.

2.1 Taligenkänning

Taligenkänning innebär automatisk omvandling av akustiska talsignaler till textsträngar. Det finns flera användningsområden för taligenkänning. Taligenkänning kan bland annat användas för diktering, automatiska telefonväxlar och som hjälpmedel för funktionshindrade. En viktig skillnad mellan dessa användningsområden är om systemet är talarberoende eller talaroberoende. Ett system som är talarberoende har möjlighet att anpassa sig till en speciell talare och kan därmed uppnå högre prestanda. Ett talaroberoende system förväntas känna igen tal av en godtycklig röst.

De största utmaningarna för taligenkänning är talets och språkets naturliga oregelbundenhet. Talet har till skillnad från skriften inga tydliga avgränsningar mellan ord. Ord kan uttalas slarvigt eller utelämnas helt. Vid samtal människor emellan finns det sällan problem med att skilja mellan bakgrundsljud, såsom sorl, buller och annat tal i bakgrunden, och tal som hör till dialogen. Detta är dock något en dator får svårt för. En annan svårighet för en dator är att utnyttja kontext för disambiguering och tolkning av grammatiska felaktigheter. Ett talaroberoende system behöver hantera olika dialekter samt variationer hos den mänskliga rösten. Även en enskild persons röst kan variera med faktorer som till exempel sinnestämning och hälsotillstånd. Röstvariationer är alltså en problematik för både talarberoendesystem och talarberoende system. Dock är det ett mycket större problem för ett talarberoende system.

De vanligaste metoderna för taligenkänning är kunskapsbaserade metoder, som till stor del bygger på lingvistisk kunskap, mönsterigenkänning, användning av neuronät samt olika typer av Markov-modellering. Dessa metoder beskrivs närmare i Holmes och Holmes (2001).

Taligenkänning kan genomföras baserat på en statistisk modell. Ett sådant system består av en akustisk modell och en språkmodell. Dessutom behövs

någon form av signalbehandling för att omvandla den akustiska signalen till en digital representation. I följande ekvationer används A för att beteckna den digitala representation som observerats.

Statistisk taligenkänning kan matematiskt beskrivas som:

$$\hat{W} = \arg \max_W P(W|A) \quad (1)$$

Där \hat{W} motsvaras av den sekvens ord som med störst sannolikhet motsvarar A . För att beräkna $P(W|A)$ kan Bayes formel användas:

$$P(W|A) = \frac{P(W)P(A|W)}{P(A)} \quad (2)$$

I Bayes formel motsvaras $P(A|W)$ av sannolikheten för att ordsekvensen W ger upphov till A . $P(A)$ motsvaras av sannolikheten för att A observeras. $P(A|W)$ och $P(A)$ beräknas av den akustiska modellen. $P(W)$ motsvaras av sannolikheten för ordsekvensen W . $P(W)$ beräknas av en språkmodell. Vid sammanslagning av (1) och (2) kommer $P(A)$ inte att påverka maximeringen av W och kan därmed tas ur ekvationen. Detta leder till följande ekvation:

$$\hat{W} = \arg \max_W P(W)P(W|A) \quad (3)$$

I den här uppsatsen kommer inte akustisk modellering och signalbehandling att diskuteras vidare eftersom det inte rör experimenten för denna uppsats. För mer information om signalbehandling och akustisk modellering se Rabiner och Juang (1993).

2.1.1 Språkmodellering

En språkmodells uppgift är att beräkna sannolikheten för en sekvens ord. Detta kan göras genom att anta att sannolikheten för ett visst ord baseras på föregående ord enligt:

$$P(W) = P(w_1^n) = \prod_{i=1}^n P(w_i|w_1^{i-1}) \quad (4)$$

Ekvation 4 visar hur sannolikheten för det första ordet multipliceras med sannolikheten för det andra ordet givet det första ordet och denna produkt multipliceras i sin tur med det tredje ordet givet de två första orden och så vidare. Språkmodellen måste alltså klara av att beräkna sannolikheter för att ett visst ord följer efter en given sekvens ord. Detta kan beskrivas:

$$P(w_i|w_1, \dots, w_{i-1}) \quad (5)$$

där w_1, \dots, w_{i-1} motsvaras av samtliga föregående ord. Dessa ord kallas för historien, medan w_i kallas för prediktionen. Det är en omöjlighet att lagra alla tänkbara historier och dessutom orimligt att anta att sannolikheten för ett ord beror på samtliga ord i historien. Därför behöver historierna grupperas. Detta kan göras genom att gruppera efter hur historierna slutar och därmed göra antagandet att två historier är ekvivalenta om de slutar på samma sätt. På så sätt kommer sannolikheten för ett visst ord att endast bero på de närmast föregående orden. Detta kallas för ett Markov-antagande. En modell som baseras på

de $n-1$ föregående orden kallas för en n -gram-modell och tillhör den $(n-1)$:a ordningen av Markov-modeller. I en n -gram-modell ska det n :te ordet förutsägas givet de $n-1$ föregående orden. Om $n=1$ kallas modellen en unigrammodell och detta är den enklaste formen av n -gram-modeller. Denna modell använder ingen kontext för att förutsäga ett visst ord. Om $n=2$ kallas modellen för en bigrammodell och sannolikheten för ett ord beror endast på det föregående ordet, $P(w_n|w_{n-1})$. Om $n=3$ kallas modellen för en trigrammodell och sannolikheten för ett ord beror på de två föregående orden, $P(w_n|w_{n-2}, w_{n-1})$.

För att estimeras dessa sannolikheter används maximum likelihood-estimering (MLE). Det innebär att man estimerar sannolikheten med den relativa frekvensen utifrån någon träningsmängd. Sannolikheten för att w_i följer efter w_{i-2}, w_{i-1} uppskattat med MLE kan skrivas som:

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \quad (6)$$

där $C(w_{i-2}, w_{i-1}, w_i)$ står för frekvensen av ordsekvensen w_{i-2}, w_{i-1}, w_i och $C(w_{i-2}, w_{i-1})$ står för frekvensen av de föregående orden.

För n -gram-modeller med högre värden på n kommer antal sedda n -gram att vara färre samtidigt som totalt möjligt antal n -gram ökar exponentiellt vid ökning av n . Detta gör att riskerna för opålitliga uppskattningar ökar då modellerna försöker uppskatta sannolikhet för n -gram som inte finns med i träningsdata och då de n -gram som finns med i träningsdata har låg frekvens.

För en modell med en vokabulär med V antal ord finns $V - 1$ oberoende variabler om modellen är en unigrammodell, eftersom sannolikheten för alla ord tillsammans ska vara lika med 1. För en bigrammodell med V antal ord finns $V^2 - 1$ antal oberoende variabler och i en trigrammodell finns $V^3 - 1$ oberoende variabler. En n -gram-modell har alltså $V^n - 1$ oberoende variabler.

Nästan oavsett hur mycket träningsdata som finns tillgänglig är det aldrig tillräckligt för att täcka in alla möjliga kombinationer av n -gram. Det kommer alltid att finnas n -gram som inte setts i träningsdata. Dessa kallas här *osedda* n -gram. Den största delen av de osedda n -grammen är kombinationer av ord som normalt inte förekommer i språket och resterande osedda n -gram har på grund av en slump inte kommit med i träningsdata. Detta fenomen kallas glesa data (sparse data). Vid användning av MLE för att estimeras sannolikheterna för ordsekvenser tas ingen hänsyn till de osedda n -grammen. Detta gör att så fort ett ord som inte förekommit i träningsdata är en av de $n-1$ föregående orden blir sannolikheten lika med noll. För att kompensera för glesa data används olika typer av utjämningsmetoder (smoothing methods), vars uppgift är att tilldela sannolikhet till de osedda n -grammen. Detta innebär samtidigt att sannolikheterna för de n -gram som finns i träningsdata måste estimeras om. En sådan metod är Laplaces regel som går ut på att lägga till 1 till samtliga frekvenser för att förhindra att sannolikheter blir noll endast på grund av att ett ord inte finns med i träningsdata. Detta ger dock för mycket sannolikhet till de osedda n -grammen i förhållande till de n -gram som faktiskt finns med i träningsdata. Lidstones regel går därför ut på att lägga till ett lägre värde till frekvenserna. En annan metod är Good-Turing-estimering. Denna estimering går ut på att omfördela sannolikheten för n -grammen efter frekvens. Endast n -gram med lägre frekvens än något tröskelvärde estimeras om och en konstant för varje frekvens tas fram. Denna konstant multipliceras sedan med nya

n -gram. För mer ingående beskrivning av dessa och andra utjämningsmetoder se Nugues (2006) eller Manning och Schütze (1999).

2.1.2 Klassbaserade n -gram-modeller

En annan metod för att hantera glesa data är att gruppera ord i klasser. Orden i klasserna hanteras som en enhet istället för enskilda ord. På så sätt minskas antal oberoende variabler i språkmodellen. Sådana språkmodeller kallas klassbaserade n -gram-modeller och det är denna typ av språkmodell som undersöks i denna uppsats. Brown m.fl. (1992) definierar en klassbaserad n -gram-modell som en n -gram-modell där

$$P(w_k|w_1^{k-1}) = P(w_k|c_k)P(c_k|c_1^{k-1}) \quad (7)$$

för $1 \leq k \leq n$. Antal oberoende variabler är $C^n - 1 + V - C$, där C är antal klasser och V är antal ord i vokabuläret. $C^n - 1$ av de oberoende variablerna kommer från n -gram-modellen och $V - C$ kommer från $P(w_i|c_i)$. $P(w_i|c_i)$ motsvarar sannolikheten för att w_k tillhör klassen c_k . En klassbaserad n -gram-modell har alltid färre oberoende variabler än motsvarande n -gram-modell utom när $V = C$ (Brown m.fl., 1992).

Minskning av antal oberoende variabler leder till en grövre modell. En grövre modell kan dock innebära att modellen förlorar värdefull information. Detta skulle kunna leda till en försämring av modellen. Det skulle till exempel kunna vara så att alla ord i klassen inte kan förekomma i exakt samma kontext. Ett exempel på det är ord med olika böjningsform som förekommer med olika artiklar.

Ord kan delas in i klasser både automatiskt och manuellt. I denna uppsats grupperas ord i klasser manuellt. För att dela in ord i klasser automatiskt kan olika typer av klustringsalgoritmer användas. Uszkoreit och Brants (2008) beskriver en metod för att klustra orden för användning i en klassbaserad språkmodell för maskinöversättning.

2.2 Kategorisering

Kategorisering innebär att en mängd objekt tilldelas fördefinierade kategorier eller klasser. Kategorisering kan även kallas klassificering. Många klassiska språkteknologiska problem såsom ordklasstagning och automatisk disambiguering av ord är exempel på kategorisering. I fallet med ordklasstagning är orden objekt och kategorierna de traditionella ordklasserna (Manning och Schütze, 1999). Kategorisering används även i applikationer för informationsökning. Inom datavetenskap används kategoriseringstekniker för att sortera bilder efter innehåll såsom landskap och ansikten (Manning m.fl., 2008). Textkategorisering innebär att dela upp en mängd dokument efter fördefinierade klasser eller kategorier. Kategorierna motsvarar ofta generella ämnen som till exempel *sport* eller *datorer*.

Statistisk textkategorisering innebär att statistiska inlärningsmetoder används för att träna en kategoriserare utifrån en träningsmängd. Träningsmängden är annoterad med korrekt kategori för respektive dokument. Detta görs ofta manuellt. Denna typ av inläring kallas övervakad inläring eftersom en

träning mängd med fördefinierade kategorier används. För övervakad träning av en kategoriserare är inlärningsmetodens uppgift att hitta en funktion γ som kan tilldela dokument kategorier: $\gamma : X \rightarrow C$ där X är mängden av dokument och C är mängden av fördefinierade kategorier. Det finns även oövervakade inlärningsmetoder som förutsätter att inlärningsmetoden själv måste hitta likheter mellan dokumenten och dela in dem i kluster. För detta kan olika typer av klustringsalgoritmer användas.

Vid kategorisering är det vanligt att ett dokument endast kan tilldelas en kategori, men det finns även kategoriserare som kan tilldela ett dokument flera kategorier. Det senare fallet kan lösas genom att träna en kategoriserare för varje kategori. Dessa kategoriserare har då till uppgift att avgöra om dokumentet tillhör kategorin eller om dokumentet inte tillhör kategorin. Detta kallas även binär kategorisering (Sebastiani, 2002).

För att kunna kategorisera dokument behövs ett lämpligt sätt att representera dessa. Ett sätt är att låta varje dokument representeras av en vektor, $\vec{w} = \langle w_1, \dots, w_{|T|} \rangle$ där T är mängden av alla ord som finns med i något av dokumenten i träningsmängden och där $w_i = 1$ innebär att ordet finns med i dokumentet och $w_i = 0$ innebär att ordet inte finns med i dokumentet. Sådana modeller kallas bag-of-word-modeller eftersom de endast tar hänsyn till vilka ord som ingår i dokumentet och inte vilken ordning orden står i. En bag-of-word-modell har heller inte någon uppfattning om vilka relationer orden har sinsemellan. Vissa modeller håller även reda på frekvenser för orden som ingår i dokumentet.

Det är rimligt att anta att alla ord inte är lika viktiga för beslutet om vilken kategori dokumenten skall höra till. Vissa grammatiska funktionsord, som till exempel *och* och *men*, kan räknas till ord som inte kommer att bidra till beslutet om vilken kategori som dokumentet ska tilldelas. Därför kan ett dokument även representeras av en viktad vektor. Viktningen motsvarar hur mycket varje ord bidrar till beslutet om vilken kategori som ska tilldelas dokumentet.

För att få en mindre komplex modell finns olika metoder för att räkna ut vilka ord i vokabulärer som bidrar mest till beslut av kategori och låta vektorn bestå enbart av index för de ord som bidrar mest till beslut om kategori. Exempel på sådana metoder är *mutual information* och χ^2 *feature selection*. För mer information om olika metoder för kategorisering och metoder för att indexera dokument se Sebastiani (2002).

2.3 Call routing

Call routing går ut på att dirigera telefonsamtal. Call routing är en kombination av två språkteknologiska problem, dels taligenkänning och dels kategorisering av resultatet av taligenkänning. Vid kategorisering motsvarar kategorierna destinationer dit samtalen kan kopplas. Det kan till exempel vara handläggare med olika kompetensområden eller automatiska telefontjänster.

Utformning av ett dialogsystem för call routing kan göras med olika metoder. Antingen kan en styrd dialog användas där användaren ger input till systemet i en bestämd ordning på en bestämd form eller så kan användaren tillåtas att ge systemet information genom att svara på en öppen fråga som till exempel *Beskriv ditt ärende* eller *Hur kan vi hjälpa dig?* Gorin m.fl. (1997) beskriver

ett dialogsystem för call routing där användaren svarar på den öppna frågan *How may I help you?* Ur resultatet av taligenkänningen extraheras betydelsen av yttrandet. Detta görs genom att försöka identifiera betydelsebärande fraser. Fraserna används sedan för att styra samtalet till rätt destination. Gorin m.fl. (1997) poängterar att det viktiga är att extrahera och identifiera de betydelsebärande fraserna och inte att lyckas känna igen varenda ord korrekt i yttrandet. Dialogsystem som använder denna typ av friare dialog kallas fritt tal-system. Det system som denna uppsats undersöker är ett fritt tal-system. Dialogsystem med fritt tal lämpar sig särskilt bra för en organisationsstruktur med många olika avdelningar, där det är svårt att förutse och täcka in allt som användarna kan tänkas säga. För taligenkänning med fritt tal används en statistisk modell i stället för en traditionell grammatik.

2.4 Utveckling av ett fritt tal-system

I detta avsnitt ges en översiktlig beskrivning av hur utvecklingsprocessen för ett dialogsystem med fritt tal kan se ut.

2.4.1 Datainsamling

För att utveckla ett dialogsystem med fritt tal som baseras på en statistisk modell behövs en stor mängd initiala data i form av yttranden. Autentiska yttranden som träningsdata är att föredra framför fiktiva yttranden. Detta kan vara problematiskt eftersom det på förhand är svårt att veta hur användarna kommer att interagera med ett system. För att få tillgång till autentiska yttranden till ett sådant system finns några olika metoder. En vanlig metod är Wizard of Oz (WOZ), som innebär att en handläggare styr dialogen via ett så kallat promptpiano. Promptpiano består av prompter som handläggaren kan välja mellan. Handläggaren bestämmer alltså vad systemet ska säga och vart samtalet ska kopplas. Dialogen spelas in och yttrandena kan användas för att träna en statistisk språkmodell. Ett sådant promptpiano som har använts för datainsamling till ett av TeliaSoneras system beskrivs i (Wirén m.fl., 2007). En annan metod för datainsamling är att spela upp en öppen fråga och sedan spela in vad användarna svarar och sedan koppla alla samtal till en och samma kö. En tredje metod är att utveckla en enkel modell som klarar av att styra de enklaste och vanligaste samtalen till rätt destination men placerar de övriga samtalen i en kö till handläggare. På så sätt får man tidigt en automatisering samtidigt som användarna inte riskerar att hamna fel.

2.4.2 Utveckling av kategoristruktur

En viktig del av utvecklingen av ett fritt tal-system är att definiera kategorier. Det är viktigt att involvera domänexperter under hela utvecklingsprocessen. Speciellt viktigt är detta vid definiering av kategorier. Boye och Wiren (2007) beskriver en metod som har använts i samtliga fritt tal-system hos TeliaSonera. Metoden innebär att två uppsättningar av kategorier definieras på olika nivåer, semantiska kategorier och applikationskategorier. De semantiska kategorierna är tänkta att representera det semantiska innehållet i yttrandet. Applikations-

kategorierna är tänkta att motsvara hur yttranden ska hanteras av systemet. De semantiska kategorierna mappas till applikationskategorierna i många-till-en-relationer. Vid en omstrukturering hos organisationen behöver endast mappningen mellan de semantiska kategorierna och applikationskategorierna göras om. På så sätt blir systemet lättare att underhålla.

Vidare beskriver Boye och Wiren (2007) en treställig kategoristruktur på formen: (*familj, intention, objekt*), där familj motsvarar produktgrupp, intention motsvarar den intention användaren har med samtalet (om det till exempel gäller beställning av produkter eller felanmälan) och objekt specificerar mer exakt vad samtalet gäller. Ett exempel på kategori hämtat från Boye och Wiren (2007) är (*mobitelefont, beställa, SIM-kort*).

I det system som undersöks i denna uppsats har kategorierna en treställig struktur liknande den som Boye och Wiren (2007) beskriver och kategorierna finns i två uppsättningar, semantiska kategorier och applikationskategorier. I denna uppsats har kategoriseraren endast tränats på applikationskategorierna.

2.4.3 Dialogstruktur

När kategoristrukturen är definierad måste även dialogstrukturen definieras, det vill säga bestämma i vilka situationer fritt tal ska tillämpas och i vilka situationer styrt tal ska tillämpas. Till definiering av dialogstruktur hör även val av prompter att spela upp. I de befintliga applikationerna i TeliaSoneras system är det endast det första yttrandet som använder fritt tal. Därefter går systemet över till styrt tal för disambiguering, bekräftelse eller för att få ytterligare information. För vissa kategorier kan systemet upprepa den öppna frågan för att ge användaren en ny chans att berätta om sitt ärende.

2.4.4 Pilot, driftsättning och underhåll

I sista fasen av produktutvecklingen behöver applikationen testas i drift i liten skala. Den första versionen som sätts i drift kallas pilotversion. Den viktigaste uppgiften för pilotversionen är att verifiera att applikationen levererar förväntat resultat i en riktig miljö med riktiga användare och handläggare. Med pilotversionen kan nya data samlas in och utvärderas. Applikationen kan sedan förbättras och byggas ut till en slutgiltig version som sätts i drift i full skala.

Under applikationens livstid kommer den att behöva kontinuerligt underhåll. En av anledningarna till det är att användarna ändrar beteende och sätt att uttrycka sig i och med att språket förändras och automatiska dialogsystem blir vanligare. Vid införande av en ny version är det vanligt att låta den nya versionen få utmana den befintliga versionen. Ett sätt är att låta de två versionerna vara i drift parallellt och styra in en mindre delmängd av alla samtal till den nya versionen. Efteråt utvärderas versionerna för att få reda på om den nya versionen kan ersätta den befintliga versionen. Detta sätt brukar kallas för *champion vs. challenger* i utvärdering av röststyrningsprojekt. Vid införande av en ny version är det lämpligt att undersöka om gruppering av ord i klasser skulle kunna förbättra applikationen.

2.4.5 Teknisk översikt av TeliaSoneras system

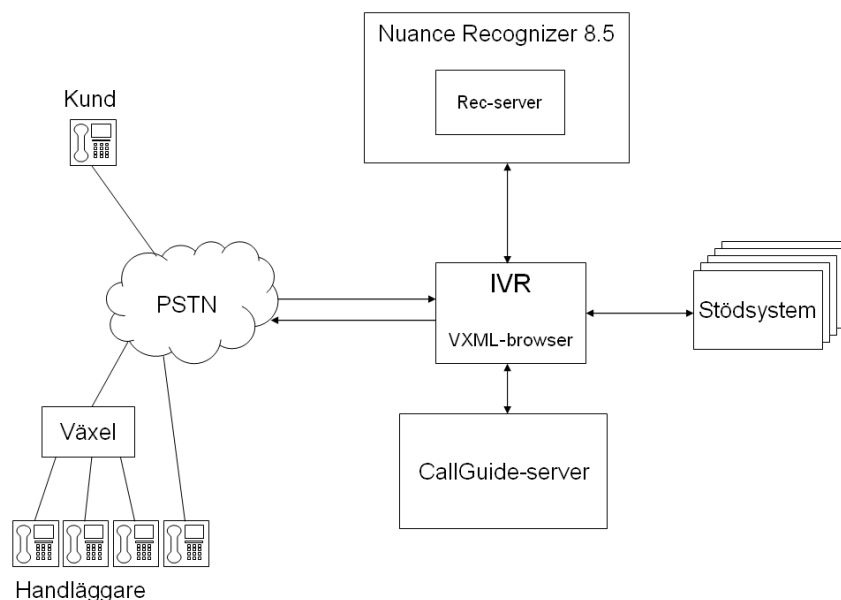
I det här kapitlet ges en övergripande bild av hur ett av systemen hos TeliaSonera ser ut. Beskrivningen utgår ifrån figur 2.1.

När en användare ringer in till systemet kopplas samtalet via det allmänna telefonnätet (PSTN, Public Switched Telephone Network) till ett talsvar (IVR, Interactive Voice Response). Ett talsvar är en applikation som interagerar med användaren och förser användaren med information. Det är vanligt att talsvaren använder tonval (DTMF) eller någon form av röststyrning för att hämta information om användarens ärende.

Talsvaret i detta system är baserat på VoiceXML. I VoiceXML-filerna definieras vilka prompter som ska spelas upp och vilka grammatiker som ska användas för taligenkänning. Talsvaret hämtar information om samtalet hos en CallGuide-server för att veta vilket VoiceXML-skript som ska köras. CallGuide är TeliaSoneras kontaktcenter-lösning. En kontaktcenter-lösning är en applikation som är tänkt att underlätta hantering av inkommande kontakter från olika kommunikationskanaler till exempel telefonsamtal, e-post, chatt, sms, mms.

Talsvaret kan behöva hämta information från något stödsystem. Det kan vara information om den som ringt till systemet, till exempel om det är en befintlig kund, prioriterad kund eller helt ny kund. När talsvaret vill ha information från användaren kan taligenkänning eller tonval användas. För taligenkänning och kategorisering används Nuance Recognizer 8.5 som består av en serverdel och en klientdel. Klientdelen som finns på talsvaret upptäcker start och slut på ljudströmmen (endpointing) och skickar sedan ljudfilen till serverdelen (Nuance recognizer) där taligenkänning genomförs.

När talsvaret har all information som behövs för vidare styrning skickas den till CallGuide som kan tala om till vilken handläggare eller självbetjäningstjänst samtalet ska styras.

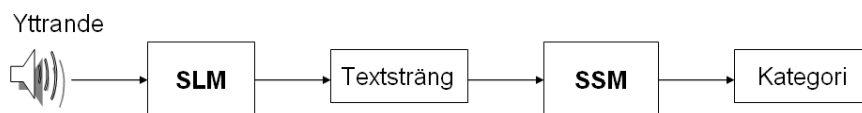


Figur 2.1: Översikt av ett av TeliaSoneras system

2.5 Nuance Recognizer 9.0

Nuance Recognizer 9.0 används för taligenkänning och kategorisering. Nuance Recognizer har stöd för taligenkänning baserad på en statistisk modell vilket är lämpligt för ett dialogsystem med fritt tal. Vanligtvis brukar man säga att dialogsystem med fritt tal inte använder någon formell grammatik utan istället använder en statistisk språkmodell. Nuance använder termen grammatik på ett mer generellt sätt för den språkmodell som används för att tolka ett yttrande oavsett om det är en formell grammatik eller en statistisk språkmodell. Termen *fritt tal-grammatik* kommer därför att användas för de fall då språkmodellen består av en statistisk språkmodell istället för en formell grammatik. I termen fritt tal-grammatik ingår även en kategoriserare som returnerar den kategori som yttrandet har tilldelats. För dialogsystem med styrt tal eller tonval kan grammatiker med GrXML-format användas (Nuance Communications, 2008b).

För uppbyggnad av en fritt tal-grammatik med Nuance Recognizer skapas en statistisk språkmodell (SLM) som returnerar en textrepresentation av yttrandet och en statistisk semantisk modell (SSM) som genomför kategoriseringen utifrån den igenkända textsträngen. Dessa två delar packas ihop till en enhet med en omslutande grammatik. Denna grammatik kompileras och genererar en binär `.gram`-fil. Denna fil kommer att innehålla både en SLM och en SSM. Det är endast denna fil som applikationen behöver tillgång till.



Figur 2.2: Call routing med Nuance Recognizer 9.0.

2.5.1 Statistisk språkmodell

Nuance Recognizer 9.0 använder en statistisk språkmodell, för taligenkänning. Träningsdatan till en SLM består av textrepresentationer av yttranden. Rekommenderat antal träningsyttranden för en vokabulär om 2 500 ord är cirka 20 000 yttranden (Nuance Communications, 2008a). Träningsfilen är en xml-fil med två element. Ett `<vocab>`-element med alla unika ord och ett `<training>`-element med alla unika yttranden samt deras frekvens. Träningsfilen kompileras med Nuance-verktyget `sgc`. Följande två filer genereras:

- en fil med suffix `.fsm` som innehåller en finit automat.
- en fil med suffix `.wordlist` som innehåller alla unika ord i modellen.

2.5.2 Statistisk semantisk modell

Med hjälp av en statistisk semantisk modell, kan resultat av taligenkänning tilldelas semantisk betydelse. Den semantiska modellen måste tränas med ett stort antal yttranden. Rekommenderat antal träningsyttranden är minst 500 yttranden per kategori (Nuance Communications, 2008a).

En SSM beräknar sannolikheten för att ett visst yttrande tillhör en viss kategori. En viktig del i skapandet av en SSM är definiering av tydliga kategorier. Mer om design av kategorier finns tidigare beskrivet i avsnitt 2.4.2.

En träningsfil för SSM består av yttranden och deras motsvarande kategorier. Träningsfilen är i XML-format. I träningsfilen är det optimalt att ange en fristående utvärderingsmängd med yttranden samt respektive korrekt kategori. Denna utvärderingsmängd används internt av Nuance för att utvärdera kategoriseraren under träning. Denna mängd kan därför ses som en del av träningen och vid slutlig utvärdering av modellen bör en annan utvärderingsmängd användas. Träningen av en SSM sker med hjälp av itereringar över träningsdatan. För varje iterering tränas systemet och utvärderas på den avskilda utvärderingsmängden. Beroende på resultatet för utvärderingsmängden upprepas detta flera gånger. I träningsfilen anger man hur många iterationer som ska genomföras. Efter varje iteration anges vilken iteration som gav bäst resultat. Resultatet för utvärderingsdatan kan skrivas till en fil. I den filen sammanfattas hur det har gått för varje kategori samt hur det gått för varje yttrande. Träningsfilen kompileras med Nuance-verktyget `ssm_train` som genererar en `.ssm`-fil som innehåller kategoriseraren.

2.5.3 Wrappergrammatik

En wrappergrammatik behövs för att koppla ihop en SLM och en SSM till en fritt tal-grammatik som kan anropas från en applikation. Grammatikfilen kommer att innehålla en SLM och en SSM och är den enda fil som behövs för att genomföra taligenkänning och kategorisering. I wrappergrammatiken anges vilka komponenter som grammatiken ska bestå av. För SLM anges `.fsm`-filen och `.wordlist`-filen. För SSM anges `.ssm`-filen. Grammatiken kompilerades med Nuance kommandot `sgc` till en binär `.gram`-fil.

Resultatet av taligenkänning och kategorisering returneras som ett XML-träd. Trädet består bland annat av element som innehåller resultat av taligenkänning, resultat av kategorisering samt konfidenser för taligenkänning och kategorisering. Applikationen kan sedan hämta element och attribut ur XML-trädet.

2.6 Varför klasser?

Som tidigare diskuterats i 2.1.2 är implementation av klasser är ett sätt att kompensera för glesa data. Liknande ord i träningsdata grupperas och orden behandlas sedan på liknande sätt av modellen. Detta är speciellt fördelaktigt för ord med låg frekvens som utan klass skulle fått lite eller ingen träningsdata.

En annan fördel med klasser är att klasserna ligger i separata grammatikfiler vilket gör att man kan lägga till och ta bort medlemmar utan att behöva lägga till nya träningsdata. Beroende på typ av klass kan detta vara en vinst i sig, även om den statistiska modellen inte skulle förbättras. Ett exempel på det är en klass med olika produktnamn i en kundtjänstdomän. Samma träningsdata kan då användas trots att nya produkter tillkommer. Det finns även stöd för att returnera ordet i klassen tillsammans med kategorin till applikationen.

Exempel på klasser i olika typer av system finns i tabell 2.1. Ord och fraser som betyder samma sak kan också användas som klasser. Klasserna blir då ett explicit sätt att uttrycka synonymi. Detta kan vara användbart för kategoriseringen om det till exempel finns många synonym-facktermer inom domänen.

Typ av system/domän	Exempel på klasser	Exempel på medlemmar
Resebokning	städer, länder, platser	Malmö, Danmark, Arlanda
Kundtjänst	produkter	namn på produkter
Fakturerings/bokningssystem	månader, datum, veckodagar	maj, tisdag

Tabell 2.1: Exempel på tänkbara klasser i olika typer av domäner.

2.6.1 Implementering av klasser

Klasserna är enkla GrXML-grammatiker där varje medlem listas. Det går även att använda en mer komplex grammatik som klass. Klassens medlemmar utgörs då av samtliga strängar som accepteras av grammatiken. Nuance Recognizer 9.0 accepterar inte GSL-grammatiker (Grammar Specification Language), men det finns ett Nuance-verktyg, `convert_gsl`, som automatiskt konverterar GSL-grammatiker till GrXML-format.

För att märka upp alla ord som tillhör en viss klass i träningsfilen för en SLM används Nuance-verktyget `tag_sf` som går igenom alla yttranden i träningsfilen och märker upp orden som tillhör en viss klass med en sökväg till GrXML-grammatiken.

I en SSM sker uppmärkningen av klasser automatiskt vid träning förutsatt att man lagt till en sökväg till GrXML-grammatiken i träningsfilen. Man behöver även ange hur modellen ska hantera klassen vid träning av SSM. Följande tre alternativ finns:

- **stem** innebär att kategorisering endast baseras på klassen. Det tas alltså ingen hänsyn till vilken medlem som påträffades.
- **fragment** innebär att kategorisering baseras på både klassens medlem och klassen.
- **remove** innebär att modellen inte tar någon hänsyn till varken klass eller medlem vid kategorisering. Detta alternativ kan användas för att skapa en så kallad stoppordlista. En stoppordlista är ett sätt att definiera ett antal ord som inte är betydelsebärande.

I wrapper-grammatiken anges vilka klasser som ska returneras till applikationen tillsammans med kategorin.

3 Metod

I det här kapitlet beskrivs tillvägagångssättet för experimenten samt de data som använts. Kapitlet inleds med avsnitt 3.1 om hur klasserna har tagits fram. Sedan följer avsnitt 3.2 om de data som använts i experimenten. Kapitlet avslutas med avsnitt 3.3 som beskriver utvärderingens design samt det testsystem som använts vid utvärdering av kategorisering med taligenkänning.

3.1 Definition av klasser

Framtagning av klasser har skett med författarens grundläggande domänkunskap. Klasserna är av olika slag och har olika funktioner för modellen. Klassernas frekvenser varierar i datamängderna. Resultat av vilka klasser som togs fram samt frekvenser för klassernas medlemmar i tränings- och utvärderingsmängd presenteras i kapitel 4. Varje klass har implementerats i två modeller, en modell med kategoriseringsalternativet fragment och en version med kategoriseringsalternativet stem. De olika kategoriseringsalternativen finns beskrivna i avsnitt 2.6.1.

3.2 Data

Applikationen som samtalen är hämtade ifrån tillhör en kundtjänstliknande domän. Användare gör förfrågningar av olika slag. Det går även att utföra olika typer av tjänster via talsvar. Det är vanligt att användare som ringer in till applikationen inte på förhand vet till vilken del av organisation deras ärende hör.

Modellerna är till största delen tränade med yttranden från våren 2007 då applikationen utvecklades. Alla yttranden är autentiska. Materialet från 2007 omfattade 17 129 manuellt transkriberade yttranden samt motsvarande ljudfiler. Rekommenderat antal yttranden som behövs för konstruering av en statistisk modell med vokabulär om 2 500 ord är cirka 20 000 (Nuance Communications, 2008a). En ny mängd yttranden samt logfiler från hösten 2008 hämtades. Denna mängd innehöll 2 128 yttranden. Dessa yttranden transkriberades och kategoriserades manuellt. Transkribering och kategorisering som görs manuellt kommer alltid att innehålla ofrånkomliga manuella fel. Transkribering beskrivs i avsnitt 3.2.1 och kategorisering beskrivs i avsnitt 3.2.2. Även den större datamängden behövde förbehandlas för att den skulle kunna användas för generering av en träningsfil. De två datamängderna blandades för att eventuella skillnader kopplade till tidpunkt på året för inspelning skulle för-

delas jämnt över tränings- och utvärderingsmängd. I avsnitt 3.2.3 beskrivs de olika datamängderna.

3.2.1 Transkribering

De nyinspelade yttranden från hösten 2008 transkriberades manuellt. För att urskilja de yttranden som använt en fritt tal-grammatik från yttranden som använt någon annan grammatik, användes ett pythonskript. Skriptet skrev ut samtliga sökvägar till ljudfiler och logfiler till yttranden som använt en fritt tal-grammatik. Skriptet hämtade även resultatet av taligenkänning och kategorisering ur respektive logfil. Detta för att underlätta vid den manuella transkriberingen och kategoriseringen. Det enda som annoterades vid transkribering var en strängrepresentation av yttrandet. Eftersom transkribering görs manuellt är det ofrånkomligt med manuella fel. Stavfel och alternativa stavningar smyger sig in i transkriberingarna trots transkriberingsregler. Detta påverkar modellens prestanda. Under arbetet med utvärderingen upptäcktes cirka ett tiotal yttranden med transkriberingsfel och stavfel i utvärderingsmängden. Dessa yttranden rättades till före den slutgiltiga utvärderingen. Det är dock högst troligt att det fortfarande förekommer fel i utvärderingsmängdens guldstandard. En guldstandard kan beskrivas som ett facit och motsvaras här av manuella transkriptioner av yttranden. Vid utvärdering är det guldstandarderna som avgör vad som anses vara korrekt respektive inkorrekt.

3.2.2 Kategorisering

Med ett pythonskript matchades yttranden mot ett lexikon innehållande tidigare manuellt kategoriserade yttranden. De yttranden som inte hittades i lexikonet kategoriserades manuellt. Kategoriseringen gjordes med grundläggande insikt i den organisationsstruktur som applikationen är tänkt för. Ett fåtal yttranden fick dock kategoriseras av domänexperter. Yttranden kategoriserades endast med applikationskategorier. Det hade varit önskvärt att de även hade kategoriserats med semantiska kategorier. Detta för att minimera risken att behöva kategorisera om yttranden ifall organisationsstrukturen skulle förändras. Även vid kategorisering är det ofrånkomligt med manuella fel. Vissa av felkategoriseringarna beror på otydligt definierade kategorier. Detta har gjort att liknande yttranden har tilldelats olika kategorier.

3.2.3 Datamängder

Den totala mängden insamlade data delades slumpvis¹ in i tre skilda mängder. Tabell 3.1 visar antal yttranden i de olika datamängderna. Test1 användes för Nuance interna utvärdering av en SSM. Test2 användes för slutlig utvärdering av modellerna. Denna mängd kallas i uppsatsen för utvärderingsmängd. Test1 och Test2 bestod av ca 10% vardera av den totala mängden yttranden. Träningsmängden bestod av ca 80% av alla yttranden och användes för träning av

¹En begränsning i slumpnings-momentet var att varje kategori skulle vara representerat i varje datamängd med minst ett yttrande. För slumpning användes pythons inbyggda random-metod.

Mängd	Antal yttranden
Test1	1925
Test2	1925
Träning	15407
TOTALT	19257

Tabell 3.1: Antal yttranden i träningsmängd och utvärderingsmängder.

modellen. Frekvensen mellan kategorierna skiljer sig kraftigt. Den största kategorin har 237 yttranden i utvärderingsmängden och de två minsta kategorierna har endast 2 yttranden. Totalt i utvärderingsmängden fanns 24 kategorier med tio eller färre yttranden, 15 kategorier med 11 till 30 yttranden, tio kategorier med 32 till 50 yttranden, sex kategorier med 50 till 100 yttranden och fyra kategorier med över 100 yttranden.

3.3 Metod för utvärdering

För utvärderingen har flera modeller med klasser skapats. Vilka klasser som skapats beskrivs i avsnitt 4.1 och vilka modeller som skapats beskrivs i avsnitt 4.2.

Utvärderingen har gjorts med både perfekt taligenkänning och med yttranden som är resultat av taligenkänning. För kategorisering med perfekt taligenkänning har kategoriseraren opererat på transkriptioner av yttranden. Här har Nuance-verktyget `parseTool` använts. Vid utvärdering har modellerna jämförts med en baseline-modell som inte har implementerat någon klass.

3.3.1 Utvärderingsmått

I detta avsnitt beskrivs de utvärderingsmått som använts i utvärderingen av taligenkänning och kategorisering.

Word Error Rate

Word Error Rate (WER) är ett mått som kan användas vid utvärdering av taligenkänning. Måttet WER försöker kvantifiera hur mycket två yttranden skiljer sig från varandra. Definitionen av WER är summan av det minsta antal insättningar, strykningar och utbyten av ord som behövs för att två yttranden ska bli lika, dividerat med antal ord i yttrandet enligt en guldstandard (Holmes och Holmes, 2001).

$$\text{WER} = \frac{S + D + I}{N}$$

S = Antal förväxlade av ord

D = Antal borttagna ord

I = Antal instoppade ord

N = Antal ord i yttrandet enligt en guldstandard

Till exempel skulle yttrandet *Jag har frågor om min faktura* som blivit igenkänt som *Jag har frågor om fakturan* få WER på $\frac{1}{3}$. I det igenkända yttrandet saknas ordet *min* och ordet *faktura* har bytts ut mot *fakturan*. Ett problem med WER är att måttet ger lika stort straff för förväxling av två helt olika ord som om endast en ändelse blivit fel. Yttrandet i exemplet ovan hade alltså haft lika hög WER om det hade blivit igenkänt som *Jag har frågor om räkningen*. WER är *Levenshtein distance* normaliserat med yttrandelängd. Levenshtein distance är ett mått som används inom bland annat maskinöversättning och språkgranskning för att beräkna hur mycket två strängar skiljer sig från varandra.

Täckning, precision och F-score

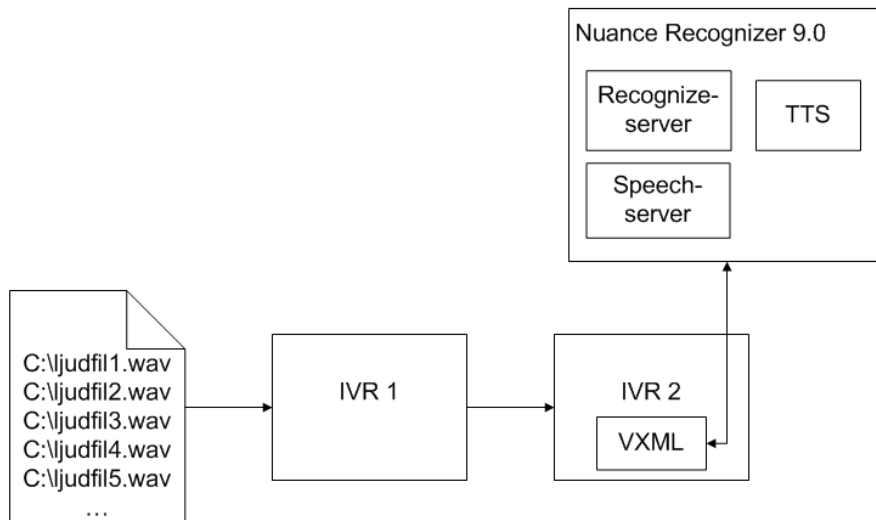
Samtliga modellers kategorisering har utvärderas med avseende på täckning, precision och F-score. Täckning, precision och F-score är tre vanligt förekommande utvärderingsmått inom flera språkteknologiska områden. För kategorisering av yttranden definieras täckning som antal korrekt kategoriserade yttranden dividerat med antal yttranden som modellen hade möjlighet att kategorisera. Precision är antal korrekt kategoriserade yttranden dividerat med antal yttranden som faktiskt kategoriserades. Precision ger ett mått på hur träffsäker modellen är. Appliceras dessa mått på kategorinivå ger täckning ett mått på hur stor andel yttranden som korrekt tilldelades en kategori givet hur många yttranden som skulle tilldelas den kategorin. Precision på kategorinivå ger ett mått på hur stor andel av de yttranden som tilldelades en viss kategori som var korrekt kategoriserade. F-score är det harmoniska medelvärdet av täckning och precision (Nugues, 2006). F-score definieras:

$$\text{F-score} = \frac{2 \times \text{Täckning} \times \text{Precision}}{\text{Täckning} + \text{Precision}}$$

För att beräkna dessa mått över en mängd kategorier finns två metoder. Makro-medelvärde (macro-average) innebär att F-score har beräknats på medelvärdet av täckning och precision för samtliga kategorier. F-score med mikro-medelvärde (micro-average) är beräknat utifrån precision och täckning för hela utvärderingsmängden. Makro-medelvärde ger alla kategorier lika stor vikt medan mikro-medelvärde ger lika stor vikt till varje yttrande (Manning och Schütze, 1999).

3.3.2 Testsystemet

För att utvärdera taligenkänningen har ett testsystem enligt figur 3.1 satts upp. Taligenkänningen har utvärderats med hjälp av två talsvar. Det ena talsvaret (IVR 1) har ringt upp det andra talsvaret (IVR 2) och spelat upp ljudfiler innehållande yttranden. Det talsvar som tar emot samtalen, IVR 2 anropar en VoiceXML-fil (VXML) som innehåller själva dialogen. Dialogen använder text-till-tal (TTS) för att spela upp en öppningsprompt. Prompten spelas upp endast för att systemet ska få lite mer tid ifall talet skulle finnas en bit in på ljudfilen. Om talet börjar tidigt avbryts prompten. Nuance Recognizer 9.0 används för taligenkänning och kategorisering. Vid taligenkänning streamas ljudet direkt till recognize-servern med hjälp av speech-servern. VoiceXML-filen loggar resultatet av taligenkänningen och kategoriseringen. Vid uppspelning av



Figur 3.1: Översikt av testsystemet som användes för utvärdering av taligenkänningen

ljudfilerna har olika mängd brus kommit med vilket gör att taligenkänningen har fått arbeta med en slumpvis mängd brus. Detta speglar dock hur systemet beter sig i drift då det finns olika mängder brus när användarna ringer från olika telefoner. Vad detta brus beror på är oklart. Med tanke på denna variation hade det varit optimalt att låta varje modell köras ett antal gånger för att kunna få fram ett medelvärde för varje modell. På grund av begränsad tid för experimentet utvärderades varje modell endast en gång.

Testsystemet och modellerna är inte optimerade, flera inställningar har kvar ursprungliga standardvärden. Det gäller till exempel tröskelvärden för att kunna skilja mellan tal och brus, tidsbegränsningar för hur långa pauser mellan ord användaren får göra, hur lång tid det får gå innan användaren börjar tala samt maximal tidslängd för ett yttrande. Andra sätt att optimera modellerna är att lägga till en konfidensmotor, eventuellt definiera egna uttalsvarianter av ord som kan läggas till i standarduttalslexikonet, märka upp sammansatta ord i träningsfiler. En konfidensmotor är ett sätt att ytterligare optimera hela modellen. En sådan tränas med ljudfiler och motsvarande kategorier. Konfidensmotorn genomför taligenkänning och kategorisering och utvärderar det internt på samma sätt som testmängden i träningsfilen. I Nuance Communications (2008a) beskrivs dessa metoder för optimering av modellen.

4 Klasser

I det här kapitlet redovisas de klasser och modeller som tagits fram för experimentet. Klasserna redovisas med en beskrivning av klassen samt frekvens i träningsdata och utvärderingsdata. Beskrivning av klasserna finns i avsnitt 4.1. Domänen som applikationen hör till är en typ av kundtjänst-domän, se avsnitt 3.2. Framtagning av klasser har skett med författarens grundläggande domänkunskap. Totalt togs sex klasser fram. Av sekretesskäl beskrivs inte exakt domän. Exakta beskrivningar av klasserna kan inte heller ges. Kapitlet avslutas med avsnitt 4.2 om vilka modeller med klasser som konstruerades för detta experiment.

Tabell 4.1 visar frekvens för yttranden med klasser i träningsdata samt frekvens för yttranden med klasser i utvärderingsdata. Tabellen visar även antal medlemmar i respektive klass.

Klass	Antal medlemmar	Antal yttranden med klass	
		Träningsdata	Utvärderingsdata
Klass1	310	74 (0,48%)	4 (0,21%)
Klass2	223	33 (0,21%)	6 (0,31%)
Klass3	12	56 (0,36%)	10 (0,52%)
Klass4	21	275 (1,78%)	41 (2,13%)
Klass5	24	534 (3,47%)	65 (3,38%)
Klass6	$> 8,6 \times 10^6$	311 (2,02%)	23 (1,19%)

Tabell 4.1: Antal medlemmar i respektive klass samt klassernas frekvens i träningsdata och utvärderingsdata.

4.1 Beskrivning av klasserna

I detta avsnitt följer en beskrivning av samtliga klasser som används i experimentet.

- **Klass1** är en av de största klasserna. Förekomsten av klassens medlemmar i både träningsdata och utvärderingsdata är mycket låg. Medlemmarna i klassen utgörs av namn av en viss typ. Det är jämförbart med namn på produkter eller fabrikat i kundtjänst-domän. Medlemmarna i klassen har tilldelats sannolikheter till stor del baserat på förekomst i träningsdata. Vissa medlemmar är betydligt mer frekventa än andra. Det är stor sannolikhet att nya medlemmar tillkommer och att de högfrekventa medlem-

marna kommer att variera med tiden. Anknytning till domän är relativt stark.

- **Klass2** har flera likheter med klass1. Klassen är stor och har många medlemmar med låg frekvens. Medlemmarna i klassen utgörs av namn på länder. Medlemmarna har även här tilldelats olika sannolikhet främst baserat på frekvens i träningsdata. Anknytning till domänen är stark för ett fåtal kategorier men inte generellt.
- **Klass3** är en generell klass med få medlemmar. I detta fall utgörs klassens medlemmar av årets månader. Eftersom det är en generell klass är anknytning till domänen inte speciellt stark. Det finns dock vissa kategorier som klassen har stark anknytning till.
- **Klass4** har stark anknytning till domänen. Klassens medlemmar är relativt frekventa och har mycket hög sannolikhet för vissa kategorier. Klassens medlemmar är synonymer.
- **Klass5** har stora likheter med klass4 då denna klass också utgörs av synonymer med stark anknytning till domänen.
- **Klass6** skiljer sig från de andra klasserna eftersom den är baserad på en komplex grammatik. Medlemmarna utgörs av precis alla strängar som accepteras av grammatiken. Klassens medlemmar har likheter med olika typer av order-id eller kund-id. Klassen har stark anknytning till domänen. När en medlem ur klassen förekommer i ett yttrande kan det vara intressant att returnera den till applikationen.

4.2 Modeller

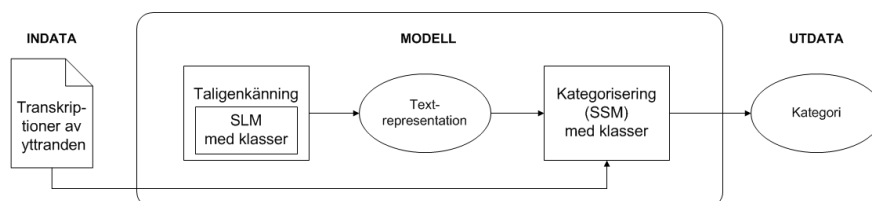
Femton modeller med klasser har skapats. Dessa modeller har utvärderats tillsammans med en baseline-modell utan klasser. Utvärderingen har skett med avseende på täckning, precision och F-score både för den totala utvärderingsmängden och för varje kategori. Modellerna bygger på de sex klasser som beskrivs i avsnitt 4.1. Varje klass har implementerats i två modeller, en med kategoriseringsalternativet *fragment* och en med kategoriseringsalternativet *stem*. Dessa alternativ beskrivs i avsnitt 2.6.1. Det har även tagits fram två modeller som implementerar samtliga klasser. Dessutom har en modell som endast implementerar klass4, klass5, klass6 skapats. Dessa klasser valdes ut för att de är bland de klasser som har starkast anknytning till domänen. Här har klass4 och klass5 använt kategoriseringsalternativet *fragment* och klass6 använt kategoriseringsalternativet *stem*. Denna modell kallas i uppsatsen för klass4-6. De andra modellerna namnges efter vilket kategoriseringsalternativ som använts samt vilken eller vilka klasser som implementerats, till exempel *stemKlass2* som motsvarar modellen där klass2 implementerats med kategoriseringsalternativet *stem*.

5 Utvärdering

I det här kapitlet beskrivs resultatet av utvärderingarna av experimenten. Kapitlet inleds med avsnitt 5.1 med utvärdering av kategorisering baserat på perfekt taligenkänning. Sedan följer avsnitt 5.2 med utvärdering av kategorisering med taligenkänning. Modellerna har utvärderats på kategorinivå och för hela utvärderingsmängden. Dessutom har kategorisering av yttranden som tilldelats en klass utvärderats för sig. I samtliga modeller har klasser implementerats i både taligenkänningen och i kategoriseringen.

5.1 Utvärdering av kategorisering med perfekt taligenkänning

Perfekt taligenkänning innebär att kategorisering har skett med transkriptioner av yttranden. Utvärderingen har utförts med Nuance-verktyget ParseTool som fick transkriptioner av yttranden som indata, se figur 5.1.



Figur 5.1: Utvärdering av kategorisering med perfekt taligenkänning

Det fanns 37 stycken yttranden som ingen av modellerna kategoriserade. Dessa yttranden består samtliga av endast ett ord. Totalt finns det 1925 yttranden i utvärderingsmängden och alltså kategoriserades 1888 yttranden. Tabell 5.1 visar täckning, precision och F-score för samtliga modeller beräknat både med makro-medelvärde och mikro-medelvärde. Modellerna är rangordnade efter F-score beräknat utifrån makro-medelvärde. Mikro-medelvärde och makro-medelvärde behandlas närmare i avsnitt 3.3.1. Modeller som korrekt har kategoriserat lika många yttranden får identiska resultat vid utvärdering med precision, täckning och F-score med mikro-medelvärde eftersom samtliga modeller har misslyckats med att kategorisera samma antal yttranden. Att precision, täckning och F-score är lägre för makro-medelvärde beror på att det finns skillnader i svårighet hos kategorierna och att de svåra kategorierna drar ner medelvärdet för alla kategorier. I tabellen syns att skillnaderna mellan modellerna är små. Samtliga modeller har korrekt kategoriserat mel-

lan 1682 och 1693 yttranden. Den modell som kategoriserat flest yttranden korrekt är klass4-6. Det är även den modell som har högst F-score med makro-medelvärde. Resultatet för fragKlass3 är identiskt med baseline-modellens resultat och resultatet för fragKlass4 är identiskt med resultatet för stemKlass4. De modeller som har färre antal korrekt kategoriserade yttranden än baseline-modellen är stemKlass2, stemKlass3 och de båda modellerna som implementerar klass1. Modellerna med kategoriseringsalternativet fragment har korrekt kategoriserat fler eller lika många yttranden som motsvarande modell med kategoriseringsalternativet stem, med undantag för modellerna som implementerar klass6 och modellerna som implementerar samtliga klasser. En anledning till att det överlag har gått bättre för fragment-modellerna skulle kunna vara att fragment-modellerna har information om vilken medlem i klassen som förekom i yttrandet och kan använda det vid kategorisering. Stem-modellerna har förlorat information om vilken medlem i klassen som förekom i yttrandet och har bara information om klassen kvar att förlita sig på vid kategorisering.

De modeller som implementerar klass6 med kategoriseringsalternativet stem verkar ha fått en fördel av att kunna behandla orden som de enskilda medlemmarna i klass6 består av som en enhet. Det är möjligt att detta underlättar vid kategorisering eftersom antalet enheter till grund för kategorisering reduceras. Att denna effekt är märkbar för modellerna som implementerar klass6 kan bero på att medlemmarna i klass6 är resultatet av en komplex grammatik och består av fler ord per medlem än medlemmarna i de andra klasserna. De tre modeller som presterade bäst både med avseende på flest korrekt kategoriserade yttranden och F-score med makro-medelvärde är de som implementerat klass6 med kategoriseringsalternativet stem.

Enligt tabellen 5.1 har fragAllaKlasser fått den näst lägsta precisionen med makro-medelvärde av alla klasser. Detta innebär att de korrekt kategoriserade yttrandena har fördelat sig jämnare över kategorierna i stemAllaKlasser än i fragAllaKlasser.

Att skillnaderna mellan modellerna är mycket små är något som är återkommande i samtliga utvärderingar. Oftast rör det sig endast om något yttrande. Om klasser verkligen gör en skillnad, så behövs det mer utvärderingsdata för att påvisa signifikans.¹

5.1.1 Kategorier som förändras med klasser i kategorisering med perfekt taligenkänning

Alla modeller har jämförts per kategori med en baseline-modell med avseende på F-score. För de flesta kategorier finns ingen skillnad. För de kategorier som har en skillnad i F-score beror den oftast på att fler yttranden har tilldelats kategorin i jämförelse med baseline-modellen vilket påverkat precisionen och därmed F-score. Eftersom skillnaderna mellan modellerna är små och det inte finns tillräckligt med data för varje kategori så tenderar skillnaden i F-score att motsvara den procentuella skillnaden som ett yttrande utgör för just den kategorin. Det handlar ofta om så små skillnader som ett yttrande. I och med att skillnaderna är så små är det orimligt att de skulle ge upphov till signifikans.

¹Inget signifikanstest har gjorts. Anledningen till det är att skillnaderna är så små att det är uppenbart att de inte kan ge upphov till statistisk signifikans.

Modell	Korr. ytrr.	Mikro-medelvärde			Makro-medelvärde		
		Prec.	Täckn.	F-score	Prec.	Täckn.	F-score
Klass4-6	1693	0,8967	0,8795	0,8880	0,8484	0,7875	0,8168
StemKlass6	1689	0,8946	0,8774	0,8859	0,8475	0,7844	0,8147
StemAllaKl.	1689	0,8946	0,8774	0,8859	0,8458	0,7855	0,8145
FragKlass4	1688	0,8941	0,8769	0,8854	0,8421	0,7859	0,8130
StemKlass4	1688	0,8941	0,8769	0,8854	0,8421	0,7859	0,8130
FragKlass2	1686	0,8930	0,8758	0,8843	0,8405	0,7865	0,8126
FragKlass6	1685	0,8925	0,8753	0,8838	0,8418	0,7844	0,8121
FragAllaKl.	1688	0,8941	0,8769	0,8854	0,8388	0,7857	0,8114
FragKlass3	1685	0,8925	0,8753	0,8838	0,8403	0,7844	0,8114
Baseline	1685	0,8925	0,8753	0,8838	0,8403	0,7844	0,8114
FragKlass5	1685	0,8925	0,8753	0,8838	0,8402	0,7841	0,8112
StemKlass3	1682	0,8909	0,8738	0,8822	0,8398	0,7833	0,8106
StemKlass2	1683	0,8914	0,8743	0,8828	0,8392	0,7832	0,8102
StemKlass5	1685	0,8925	0,8753	0,8838	0,8374	0,7845	0,8101
StemKlass1	1683	0,8914	0,8743	0,8828	0,8398	0,7819	0,8098
FragKlass1	1683	0,8914	0,8743	0,8828	0,8390	0,7819	0,8094

Tabell 5.1: Samtliga modellers totalt antal korrekt kategoriserade yttranden (Korr. ytrr.) för hela utvärderingsmängden samt precision (Prec.), täckning (Täckn.) och F-score med både makro-medelvärde och mikro-medelvärde. Tabellen är sorterad efter F-score för makro-medelvärde.

Det skiljer mycket i antal påverkade kategorier mellan modellerna. De tre modeller som presterat bäst enligt tabell 5.1 hade även påverkat flest kategorier.

5.1.2 Yttranden med klasser

Tabell 5.2 visar resultatet för de yttranden som hade tilldelats en klass. Samtliga yttranden som skulle ha en klass hade tilldelats en klass eftersom kategorisering skett med transkriptioner av yttranden. Dessa yttranden fick parsas ytterligare en gång med baseline-modellen för att utvärdera hur yttrandena hade påverkats av implementering av klasser. Tabellen visar hur många yttranden som skulle tilldelas en klass i respektive modell samt hur många yttranden som kategoriserats korrekt i modellen med klass och i baseline-modellen. I tabellen visas även korrekthet för kategoriseringen samt skillnad i korrekthet mellan baseline-modellen och modellen med klass.

För frag-modellerna finns en liten men stabil förbättring. Ingen av frag-modellerna har fått en negativ skillnad. För stem-modellerna däremot finns både positiva och negativa skillnader. Detta tyder på att för yttrandena med klass har frag-modellerna fått en fördel av att behålla information om vilket ord som fanns med i yttrandet. I denna tabell liksom i tabellerna ovan är skillnaderna dock små och handlar ofta om enstaka yttranden. Fler yttranden med klasser i utvärderingsdata skulle behövas för att få signifikanta skillnader. Resultaten tyder på att det finns anledning att tro att klasserna har en positiv inverkan på modellen även för de yttranden som inte innehåller klasser. Exempelvis har modell klass4-6, enligt tabell 5.1, korrekt kategoriserat åtta yttranden fler än

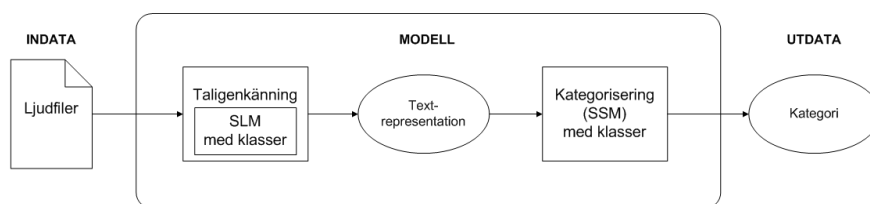
baseline-modellen men endast två av dessa yttranden kan förklaras med en förbättring för yttranden med klasser, enligt tabell 5.2.

Modell	Yttr. m. klass	Korr. yttr.		Korrekthet		Skillnad
		Baseline	Klass	Baseline	Klass	
FragKlass1	4	4	4	1,00	1,00	0,00
FragKlass2	6	3	4	0,50	0,67	0,17
FragKlass3	10	5	5	0,50	0,50	0,00
FragKlass4	41	33	34	0,80	0,83	0,02
FragKlass5	65	60	60	0,92	0,92	0,00
FragKlass6	23	17	17	0,74	0,74	0,00
FragAllaKlasser	142	116	116	0,82	0,82	0,00
StemKlass1	4	4	4	1,00	1,00	0,00
StemKlass2	6	3	2	0,50	0,33	-0,17
StemKlass3	10	5	3	0,50	0,30	-0,20
StemKlass4	41	33	34	0,85	0,87	0,03
StemKlass5	65	60	59	0,92	0,91	-0,02
StemKlass6	23	17	17	0,74	0,74	0,00
StemAllaKlasser	142	116	116	0,82	0,82	0,00
Klass4-6	127	108	110	0,85	0,87	0,02

Tabell 5.2: Kategorisering med perfekt taligenkänning av yttranden med klasser (Yttr. m. klass).

5.2 Utvärdering av kategorisering med taligenkänning

I det här avsnittet beskrivs utvärderingen av kategorisering med taligenkänning, se figur 5.2. För utvärdering har det testsystem som beskrivs i avsnitt 3.3.2 använts.



Figur 5.2: Utvärdering av kategorisering med taligenkänning

Tabell 5.3 visar samtliga modellers WER för taligenkänning och resultat av kategorisering med taligenkänning. I tabellen finns antal korrekt kategoriserade yttranden, totalt antal kategoriserade yttranden samt F-score för respektive modell. Utvärderingsmängden om 1925 yttranden är densamma som för tidigare experiment. Antal yttranden som kategoriserades varierade mellan 1867 och 1890. Alla yttranden som fick en textrepresentation har kategoriserats.

Dock finns det vissa yttranden som av någon anledning inte fick en textrepresentation. Det kan bero på att systemet antingen ansåg att det inte fanns något tal på ljudfilen eller att modellen inte lyckades känna igen det tal som den ansåg fanns på ljudfilen. Dessa problem kan bero på den slumpvisa mängd brus som förekommit i det testsystem som har använts. Att modellen lyckas identifiera tal men inte lyckas tilldela en textrepresentation ska inte kunna hända i ett system med fritt tal i drift. Modellen ska alltid tilldela yttranden den mest sannolika textrepresentationen om tal kan identifieras. Problemet med att identifiera tal beror förmodligen på inställningar för vilka ljud som motsvarar tal och vilka ljud som motsvarar brus. Systemet är inte perfekt inställt och många inställningar, till exempel konfidensnivåer har kvar ursprungliga inställningar och därmed sällas vissa yttranden bort på grund av att de av systemet ansetts vara brus. Ett alternativ hade varit att plocka bort dessa yttranden ur utvärderingen. För att kunna göra detta på ett bra sätt hade det varit optimalt att köra varje modell flera gånger som diskuterades i avsnitt 3.3.2. En tredje grupp av yttranden som inte har fått en textrepresentation i taligenkänningen är yttranden som av någon anledning har fått någon typ av tekniskt fel. Eftersom den största delen av ljudfilerna inte tidigare har gått genom taligenkänning skulle detta kunna bero på fel i ljudfilerna. Det kan även bero på något fel i det testsystem som använts.

Tabell 5.3 visar en korrelation mellan WER och F-score. Med undantag för stemKlass1 och fragKlass5, har en lägre WER korrelerat med förbättrad F-score. Detta tyder på att en förbättring av taligenkänningen även ger en förbättring av kategoriseringen. Detta är dock ingen självklar korrelation.

StemKlass1 har fått en förbättring av F-score eftersom antal korrekt kategoriserade yttranden har förbättrats. Detta trots en försämring av WER. FragKlass5 har fått en förbättrad F-score trots att antal korrekt kategoriserade yttranden är lägre än antal korrekt kategoriserade yttranden med baseline-modellen. Detta beror på att modellen har fått en högre precision eftersom totalt antal kategoriserade yttranden var lägre för fragKlass5 än för baseline-modellen.

De fyra modeller som har presterat bäst med avseende på F-score var fragKlass1, fragKlass2, stemKlass1 samt stemKlass2. Dessa modeller är inte bland de modeller som presterade bäst i utvärderingen med perfekt taligenkänning i avsnitt 5.1. Det tyder på att olika klasser är lämpliga i taligenkänningen och i kategoriseringen. Värt att notera är att klass1 och klass2 är de största klasserna om man bortser från klass6 som är en komplex grammatik och därmed har flest medlemmar.

I denna utvärdering blir kritiken till utvärderingsmättet WER påtaglig. En stor del av de insamlade yttrandena innehåller endast ett ord och skulle detta bli fel på grund av förväxlingar av ändelser och böjningsform ger WER lika hårt straff som om det hade förväxlats med ett helt annat ord. (Se bakgrundskapitlet om WER i avsnitt 3.3.1.) Det har i efterhand upptäckts fel i språkmodellens vokabulär. När felstavade ord förekommer i träningsdatan hamnar felstavade ord i språkmodellens vokabulär. Det gäller även ord och fraser som är inkonsekvent transkriberade. Ett försök att kvantifiera dessa manuella missar har gjorts genom undersökning av språkmodellens vokabulär. Vid genomgång av de 3 199 unika ord som ingår i språkmodellen hittades ett 50-tal ord som antingen var felstavade eller inkonsekvent transkriberade och därmed inte borde ha före-

kommit i modellens vokabulär. Vilken frekvens dessa ord har i träningsdatan är okänt men för de fall då det rör sig om stavfel är det sannolikt att frekvensen är låg. De manuella missarna i träningsdatan upptäcktes inte innan modellen tränades och är därför en felkälla för experimentet.

Modell	WER	Korr. yttr.	Totalt kat. yttr.	F-score
StemKlass2	0,3442	1522	1881	0,7998
StemKlass1	0,3490	1510	1880	0,7937
FragKlass2	0,3414	1509	1890	0,7911
FragKlass1	0,3450	1506	1887	0,7901
FragKlass5	0,3586	1504	1886	0,7893
Baseline	0,3458	1505	1890	0,7890
FragKlass6	0,3499	1498	1881	0,7872
StemKlass5	0,3558	1500	1886	0,7872
StemKlass4	0,3464	1500	1888	0,7868
Klass4-6	0,3574	1494	1873	0,7867
StemKlass3	0,3471	1496	1880	0,7863
FragKlass3	0,3508	1491	1878	0,7841
StemKlass6	0,3465	1487	1870	0,7837
StemAllaKlasser	0,3567	1489	1880	0,7827
FragKlass4	0,3652	1485	1877	0,7812
FragAllaKlasser	0,3991	1406	1867	0,7416

Tabell 5.3: Samtliga modellers WER, antal korrekt kategoriserade yttranden (Korr. yttr.), totalt antal kategoriserade yttranden (Totalt kat. yttr.) och F-score. Tabellen är sorterad efter F-score.

5.2.1 Kategorier som förändras med klasser i taligenkänning och kategorisering

WER och kategorisering har även undersökts per kategori för varje modell. De kategorier som har undersökts är de tio kategorier med störst förbättring av WER och de tio kategorier med störst försämring av WER. Anledningen till detta urval är att det är störst sannolikhet att det finns signifikanta skillnader för de kategorier som skiljer sig mest från baseline-modellen. Dock borde denna metod ha kompletterats med en frekvenströskel för att inte de små kategorierna skulle bli överrepresenterade i utvärderingen. Även i denna utvärdering är skillnaderna små och handlar oftast om enstaka yttranden. På grund av för få yttranden per kategori är det svårt att dra några definitiva slutsatser.

Som konstaterades och diskuterades i avsnitt 5.2 har olika modeller kategoriserat olika antal yttranden. Denna felkälla påverkar utvärderingen på kategori nivå i stor utsträckning eftersom frekvenserna för kategorierna redan är låg och WER tenderar att bli lägre för få yttranden.

För de flesta kategorier har en förbättring av WER korrelerat med en förbättring av F-score. För modell stemKlass2 som har högst F-score enligt tabell 5.3, har även större delen av de kategorier som fått en försämring av WER fått

en förbättring av F-score. Modell `fragAllaKlasser` som är en av de modeller som presterade sämst har inte fått någon förbättring av F-score för någon av de 20 kategorier som har störst förändring av WER.

5.2.2 Yttranden med klasser

I detta avsnitt undersöks de yttranden som har tilldelats en klass. Flera av de yttranden som tilldelats en klass skulle inte ha någon klass och vice versa.

Även i denna utvärdering är skillnaderna små och frekvensen för yttranden med klasser i utvärderingsdata är låg.

I tabell 5.4 visas antal yttranden som tilldelats en klass. Tabellen delar upp dessa yttranden i två grupper. En grupp för de yttranden som korrekt tilldelades en klass och en grupp för de yttranden som felaktigt tilldelades en klass. För varje grupp visas hur många yttranden som blivit korrekt kategoriserade samt WER för de korrekt kategoriserade yttrandena. Eftersom alla modeller inte har kategoriserat lika många yttranden, har de yttranden som inte blivit korrekt kategoriserade inte nödvändigtvis blivit felkategoriserade.

För de yttranden som har tilldelats en klass och som enligt guldstandarderna ska ha en klass syns en förbättring. Modellerna med klasser har genomgående presterat bättre eller lika bra. Detta tyder på att klasserna hjälper de yttranden som de är tänkta att förbättra. För de yttranden som har tilldelats en klass men som inte skulle ha en klass har modellen med klass ibland presterat bättre och ibland sämre. Tabellen visar även att korrektheten för kategorisering av dessa yttranden är låg. Detta tyder på att det är relevant att minimera antal yttranden som tilldelas klasser felaktigt. Med bättre taligenkänning skulle kanske dessa yttranden kännas igen korrekt och då inte tilldelas en klass. En felkälla för `klass4` är att det finns ord som borde varit medlemmar i klassen men som har missats vid konstruering av klassen. Det förekommer att ord i yttranden har blivit igenkända som medlemmar i en klass och därför korrekt tilldelats en klass trots att det enligt guldstandarderna inte skulle ha någon.

För `stemKlass2` som enligt tabell 5.3 hade 17 korrekt kategoriserade yttranden mer än baseline-modellen, hade endast två yttranden tilldelats en klass korrekt. Både baseline-modellen och `stemKlass2` hade kategoriserat ett av dessa yttranden korrekt. Detta är ytterligare ett exempel på att klasserna verkar göra skillnad för hela modellen.

De modeller som implementerar `klass1` respektive `klass6` har svårigheter att känna igen klassens medlemmar. Ingen av modellerna som enbart implementerar `klass1` lyckades tilldela något yttrande en klass. Modellerna som implementerar `klass6` har endast tilldelat två till tre yttranden en klass. Totalt i utvärderingsdata fanns 23 stycken yttranden med medlemmar i `klass6`. De modeller som implementerar `klass6` var bland de som presterade bäst i utvärderingen med perfekt taligenkänning. Detta skulle kunna innebära att det är rimligt att vänta med denna klass i taligenkänningen men behålla den i kategoriseringen. Grammatiken som klassens medlemmar utgörs av är komplex och det är vanligt att endast delar av klassens medlemmar känns igen korrekt, vilket leder till att yttrandet inte tilldelas någon klass. Det är möjligt att modifieringar av grammatiken så att den blir mer accepterande skulle göra att fler yttranden som innehåller en klass även tilldelas en klass.

Modell	Totalt antal yttranden som tilldelats klasser	Antal yttranden som skulle tilldelas en klass enl. guldstandard	Baseline		Klass		Antal yttranden som inte skulle ha en klass		Baseline		Klass	
			Kat.	WER	Kat.	WER	Kat.	WER	Kat.	WER	Kat.	WER
fragKlass1	0	0	-	-	-	-	-	-	-	-	-	-
fragKlass2	2	2	1	0,1250	1	0,1250	0	0,0000	0	0,0000	0	0,0000
fragKlass3	3	3	1	0,2222	2	0,3943	0	0,0000	0	0,0000	0	0,0000
fragKlass4	42	34	30	0,2029	30	0,2056	8	0,6667	1	0,6667	1	1,0000
fragKlass5	54	46	38	0,1893	39	0,3750	8	0,0500	6	0,0500	3	0,8878
fragKlass6	3	2	1	0,0952	2	0,2857	1	0,0000	0	0,0000	1	0,5294
fragAllaKlasser	98	83	68	0,1696	72	0,2775	15	1,0722	6	1,0722	7	0,6929
stemKlass1	0	0	-	-	-	-	-	-	-	-	-	-
stemKlass2	3	2	1	0,1250	1	0,3125	1	0,0000	0	0,0000	1	0,6809
stemKlass3	2	2	2	0,5370	2	0,5740	0	0,0000	0	0,0000	0	0,0000
stemKlass4	37	29	27	0,1483	27	0,1296	8	0,8333	2	0,8333	2	1,0000
stemKlass5	54	48	41	0,1602	44	0,3881	6	0,8667	3	0,8667	3	0,8667
stemKlass6	2	1	0	0,0000	1	0,1875	1	0,0000	0	0,0000	1	0,4118
stemAllaKlasser	109	89	74	0,1752	78	0,3389	20	0,8960	10	0,8960	7	0,9769
klass4-6	95	80	67	0,1605	72	0,3136	15	0,9185	9	0,9185	8	0,9604

Tabell 5.4: Samtliga yttranden som tilldelades klasser i taligenkänningen. Kolumnen totalt antal yttranden som tilldelats en klass är summan av kolumnerna Antal yttranden som skulle tilldelas en klass enl. guldstandard och Antal yttranden som inte skulle ha en klass.

6 Slutsats

I detta avslutande kapitel sammanfattas resultatet av experimenten i avsnitt 6.1 och i avsnitt 6.2 diskuteras hur vidare forskning inom ämnet skulle kunna se ut.

6.1 Resultat

Resultaten ger inte något direkt stöd för slutsatsen att klasser bidrar till en förbättrad modell. Resultaten ger inte heller något stöd för slutsatsen att klasser inte bidrar till en förbättrad modell. Det är dock möjligt att en studie med mer data samt dessa eller andra klasser skulle kunna påvisa en positiv effekt.

6.1.1 Kategorisering med perfekt taligenkänning

Utvärderingen av kategorisering med perfekt taligenkänning visar att skillnaderna mellan modellerna är små, endast ett tiotal yttranden skiljer mellan den bästa och den sämsta modellen. Resultatet tyder på att det finns skillnader mellan kategoriseringsalternativen stem och fragment. Både i utvärderingen av hela utvärderingsmängden och i utvärderingen av enbart yttranden med klasser har fragment-modellerna presterat bättre än stem-modellerna, med undantag för modellerna som implementerade klass6. En anledning till det skulle kunna vara att med kategoriseringsalternativet fragment har modellen kvar information om vilken medlem som fanns med i yttrandet och kan använda sig av det vid kategoriseringen tillsammans med klassen. Med kategoriseringsalternativet stem har modellen endast klassen att förlita sig på vid kategorisering. För de modeller som implementerar klass6 har stem-modellerna presterat bättre än motsvarande modell med kategoriseringsalternativet fragment. En hypotes till detta är att kategoriseringsalternativet stem förstör alla samband som skulle kunna uppstå mellan klassens medlemmar och olika kategorier. Utvärderingen tyder på att modellerna som implementerat klass6 med kategoriseringsalternativet stem har fått en fördel av detta. De modeller som presterade bäst i utvärderingen av kategorisering med perfekt taligenkänning var de modeller som implementerat klass6 med kategoriseringsalternativet stem.

6.1.2 Kategorisering av resultatet av taligenkänning

Utvärderingen av kategoriseringen med taligenkänning och WER visar en korrelation mellan WER och F-score. Detta är ingen självklar korrelation eftersom en förbättring i taligenkänningen inte nödvändigtvis leder till en förbättrad

kategorisering. Att det är olika modeller som presterade bäst i denna utvärdering och i utvärderingen med perfekt taligenkänning tyder på att det finns grund att tro att modeller med olika klasser i taligenkänning och kategorisering är befogade. De modeller som har fått en förbättring i WER jämfört med baseline-modellen är stemKlass1 och modellerna som implementerade klass2. Dessa modeller har även en förbättring av F-score. Tillsammans med fragKlass1 är dessa modeller de enda som har fler korrekt kategoriserade yttranden än baseline-modellen. I denna utvärdering har olika modeller kategoriserat olika antal yttranden. Denna felkälla har påverkat hela utvärderingen, speciellt påtagligt är det för utvärderingen på kategorinivå eftersom frekvenserna för kategorierna redan är låg.

Överlag verkar de kategorier som har fått en förbättring av WER även fått en förbättring av F-score. Det är dock tydligt att det behövs fler yttranden per kategori för att få signifikanta skillnader.

6.1.3 Klasser

I detta experiment utvärderades sex klasser som implementerades i 15 modeller. Klasserna var av olika slag och visade sig förbättra olika moment. Resultatet tyder på att de stora klasserna, klass1 och klass2, förbättrar taligenkänningen och de klasser som förbättrar kategoriseringen är klass4 och klass6.

Frekvens för klassernas medlemmar i utvärderingsdata är låg. Tre av klasserna har endast tio eller färre yttranden med klass i utvärderingsdata. Det behövs alltså göras fler utvärderingar med fler yttranden med klasser för att undersöka klassernas påverkan på modellen.

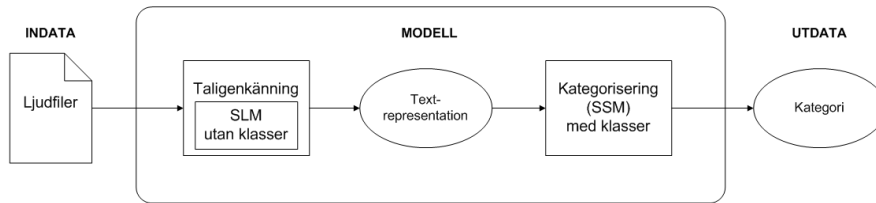
Medlemmarna i de stora klasserna hade svårt att bli igenkända trots det fick modellerna som implementerade dessa klasser ett bättre resultat än baseline-modellen i experimentet med taligenkänning. Detta tyder på att klasser även gör skillnad för yttranden som inte innehåller någon klass. Klass6 vars medlemmar utgörs av en komplex grammatik hade svårigheter att bli igenkända men bidrog till förbättring i kategoriseringen, vilket tyder på att det är lämpligt att endast ha med klassen i kategoriseringen i väntan på att taligenkänningen blir bättre och klarar av att identifiera klassens medlemmar. Vid taligenkänning finns en risk att ord som inte är medlemmar i någon klass tolkas som medlem i en klass och därför kategoriseras fel. Det är därför viktigt att minimera antal yttranden som felaktigt tilldelas klasser.

En annan aspekt av klasser i Nuance Recognizer 9.0 är att systemet blir lättare att underhålla eftersom medlemmar kan läggas till och tas bort utan att ny träningsdata behöver läggas till. Detta innebär att om det visar sig att klasser inte gör någon skillnad för modellen kan det ändå finnas fördelar med att implementera klasser för att underlätta framtida förvaltning.

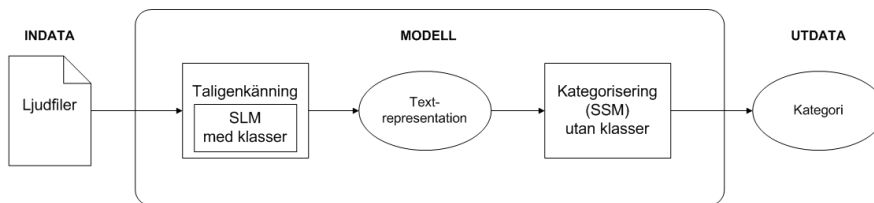
6.2 Fortsatt arbete

För att ta reda på om de observerade skillnaderna är signifikanta behöver experimenten göras om med större utvärderingsmängd och med fler yttranden per kategori. Det behövs även fler yttranden med klasser i utvärderingsdata.

Experiment där kategorisering tillämpas på resultaten av taligenkänning har endast undersökt klasser i både taligenkänning och i kategorisering. Det vore intressant att undersöka klasser endast i kategoriseringen, se figur 6.1 samt klasser endast i taligenkänningen, se figur 6.2.



Figur 6.1: Tänkbar utvärdering av modell som implementerar klasser endast i kategoriseringen.



Figur 6.2: Tänkbar utvärdering av modell som implementerar klasser endast i den statistiska språkmodellen i taligenkänningen.

Enligt resultatet är det relevant att undersöka modeller som implementerar olika klasser i taligenkänningen och i kategoriseringen. Det är även relevant att undersöka andra klasser och andra kombinationer av klasser. Exempel på klasser att undersöka är andra typer av namnentiteter. För att förbättra kategoriseraren skulle man kunna skapa en så kallad stoppordlista med hjälp av kategoriseringsalternativet *remove* som gör att kategoriseraren varken tar hänsyn till klassens medlem eller klassen.

Ytterligare ett sätt att vidare undersöka klasser är att införa dem i drift och göra löpande utvärderingar för att se hur klasserna påverkar kategoriseringen. Detta skulle medföra de praktiska fördelar som klasser har ur ett förvaltningsperspektiv.

Litteraturförteckning

- Boye, Johan och Wiren, Mats. Multi-slot semantics for natural-language call routing systems. I: *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, ss 68–75, Rochester, NY, April 2007. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W/W07/W07-0310>.
- Brown, Peter F., Della Pietra, Vincent J., Desouza, Peter V., Lai, Jennifer C., och Mercer, Robert L. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
- Gorin, A. L., Riccardi, G., och Wright, J. H. How May I Help You? *Speech Communication*, 23:113–127, October 1997.
- Holmes, John och Holmes, Wendy. *Speech Synthesis and Recognition*. Taylor & Francis, andra utgåvan, 2001.
- Manning, Christopher D., Raghavan, Prabhakar, och Schütze, Hinrich. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- Manning, Christopher D. och Schütze, Hinrich. *Foundations of statistical natural language processing*. The MIT press, 1999.
- Nuance Communications, Inc. *Nuance Recognizer 9.0 Grammar Developer's Guide*, november 2008a.
- Nuance Communications, Inc. *Nuance Recognizer 9.0 Migration for Nuance 8.5*, november 2008b.
- Nugues, Pierre M. *An Introduction to Language Processing with Perl and Prolog: An Outline of Theories, Implementation, and Application with Special Consideration of English, French, and German*. Springer, 2006.
- Rabiner, Lawrence och Juang, Biing-Hwang. *Fundamentals of speech recognition*. Prentice Hall PTR, 1993.
- Sebastiani, Fabrizio. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.
- Uszkoreit, Jakob och Brants, Thorsten. Distributed word clustering for large scale class-based language modeling in machine translation. I: *Proceedings of ACL-08: HLT*, ss 755–762, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P/P08/P08-1086>.

Wirén, Mats, Eklund, Robert, Engberg, Fredrik, och Westermark, Johan. Experiences of an in-service wizard-of-oz data collection for the deployment of a call-routing application. I: *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, ss 56–63, Rochester, NY, April 2007. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W/W07/W07-0308>.