

Textkategorisering med Predikat-Argument-Strukturer

Jacob Persson

Examensarbete för 30 hp
Institutionen för datavetenskap, Naturvetenskapliga fakulteten, Lunds universitet

Thesis for a diploma in computer science, 30 ECTS credits
Department of computer science, Faculty of science, Lund university

Sammanfattning

Textkategorisering med Predikat-Argument-Strukturer

De flesta metoder för textkategorisering använder sig av vektorrymdmodellen tillsammans med en säck-med-ord representation. Som namnet antyder ignorerar representationen meningsstrukturen i texten och tar endast hänsyn till isolerade ord. I denna uppsats undersöker jag om semantiska och syntaktiska egenskaper förbättrar precisionen i text-kategorisering. Jag utför experiment med tre semantiska och syntaktiska utökningar: ordbetydelse, subjekt-verb-objekt tripplar och rollsemantiska predikat-argument-tupler och jämför deras bidrag mot en standardliserad baslinje av säck-med-ord på Reuters korpus (RCV1). Experimenten visar att tillägget av de här representationerna ger en felminskning på 2-10 procent för kategorier som innehåller mer än 2000 träningsexempel.

Abstract

Text Categorization using Predicate-Argument Structures

Most methods of text categorization use the vector space model together with a bag-of-words representation. As the name suggests, bag of words ignores the structure of the text and only takes into account isolated unrelated words. In this thesis, I investigate whether semantic and syntactic features can improve the accuracy of text categorization. I conducted experiments on three semantic and syntactic extensions: word senses, subject-verb-object triples, and role semantic predicate-argument tuples and compared their contribution against a standard baseline of bag of words on the Reuters corpus (RCV1). The experiments showed that the contribution of these features reduces the error of the bag of words baseline by 2-10 percent on categories with more than 2000 documents in the training set.

Förord

Jag vill främst tacka min handledare Pierre Nugues samt Richard Johansson. Pierre gav många bra råd för hur jag skulle gå till väga med arbetet och Richard gjorde sin semantiska parser tillgänglig som ligger till grund för arbetet. Jag vill också tacka Pierre och Richard för artikeln vi skrev tillsammans vilket jag lärde mig mycket av samt att den lade grunden för denna uppsatsen. Till sist vill jag tacka alla vänner och bekanta som hjälpt till att korrekturläsa uppsatsen.

Innehåll

1	Inledning	3
1.1	Introduktion till textkategorisering	3
1.2	Målet med mitt arbete	4
1.3	Användningsområden	5
2	Representationer av text för automatiskt klassificering	7
2.1	Vektorrymdsmodellen	7
2.2	Ordbaserad representation	7
2.3	Utöka en representation med semantiska egenskaper	8
3	Automatisk identifiering av semantiska roller	10
4	Algoritmer för klassificering	12
4.1	Supportvektormaskiner	12
4.2	K-nn	13
4.3	Rocchio	13
4.4	Naive Bayes	13
5	Experimentuppställning	15
5.1	Korpus	15
5.2	Klassificeringsmetod	15
5.3	Korpustagging och parsers	16
5.4	Egenskapsmängder	16
6	Resultat	18
6.1	Evalueringsmetod	18
6.2	Resultat	19
6.3	Slutsats	20
A	Ordlista	24

Kapitel 1

Inledning

1.1 Introduktion till textkategorisering

När man pratar om kategorisering av text finns det två olika scenarion. I det första scenariot har man enbart texterna man vill kategorisera tillgängliga som data; man vet alltså inte vilka kategorier som är relevanta. Det här scenariot kallas för klustring då man inte stämplar texterna med kategorier utan grupperar texterna i grupper av likartade texter. Det andra scenariot som är det jag undersöker är när man har en samling texter som är utmärkta med vilka kategorier de tillhör. Målet är att träna ett system på de handkategoriserade texterna så att det automatiskt kan tilldela nya texter de kategorier som har blivit identifierade bland de handkategoriserade texterna.

Nedan följer två texter från Reuters korpus samt de kategorierna de har blivit tilldelade. Texterna har först blivit automatiskt kategoriserade sedan har en människa verifierat dem och eventuellt gjort ändringar, Kategorierna ingår i en hierarki, t.ex. är C152 en underkategori till C15 som i sin tur är en underkategori till CCAT.

INTERVIEW - Orange says growth misunderstood

UK mobile telephone operator Orange Plc, asked to comment on reasons why its share price has fallen below its flotation price, said the extent of continued growth in the British market was not fully understood.

"A lot of people do not understand what this transition process in our marketplace is all about", Orange group managing director Hans Snook said in an interview, referring to concern that growth in the UK market was slowing down.

"In fact, our net growth has increased dramatically...all the key drivers of the business, we are succeeding on", he said.

- Industries: I79020 (Telecommunications)
- Topics: C15 (corporate and industrial performance), C152 (forecasts, performance reviews, comment and recommendations, investment research), CCAT (all corporate-Industrial)

- Regions: UK

SFX Broadcasting to buy four FM stations

SFX Broadcasting Inc said on Tuesday it has signed a binding agreement to acquire ABS Communications LLC, which will own four Richmond, Va., FM radio stations, for \$37.5 million

ABS owns two stations in Richmond and also has rights to buy two additional stations, said SFX, which already owns one FM station in the Richmond market.

SFX said it will acquire WVGO(FM) and WLEE(FM) for \$14.5 million, the company said. Simultaneously with this deal, an SFX unit will acquire WKHK(FM) and WBZU(FM) from ABS Communications for \$23.0 million.

Ken Brown, president of ABS, and Ed Conrad, chief financial officer of ABS, will continue to hold a small equity stake in the group of stations and will manage them, SFX said.

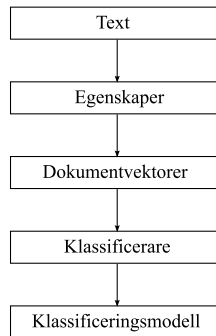
- Industries: I9741105 (Radio Broadcasting)
- Topics: C18 (all changes of ownership), C181 (acquisitions, divestments, mergers, buy-outs, buy-ins, share stakes), CCAT (all corporate-Industrial)
- Regions: USA

För att en dator automatiskt ska kunna kategorisera texterna måste texterna överföras till en representation där texters likhet kan mätas. En sådan representation är vektorrymdmodellen (beskrivs i Sek 2.1) där texterna representeras av dokumentvektorer. Vektorn anger vilka egenskaper en text har, vilket ofta är synonymt med vilka ord texten innehåller.

Statistisk textkategorisering kan delas upp i fem steg som visas i Figur 1.1. I första steget plockas egenskaper ut ur texterna som ska användas till att representera texterna. Därefter konstrueras dokumentvektorer utifrån egenskaperna där platserna i vektorerna indikerar förekomsten av en egenskap i texten. Tredje steget är att mata in dokumentvektorerna i en klassificerare som tränas till att kunna avgöra vilka kategorier en dokumentvektor tillhör. I det sista steget sparas den tränande klassificeraren i en klassificerings modell. Modellen används sedan för att klassificera nya texter genom att genom utföra de tre första stegen på de nya texterna.

1.2 Målet med mitt arbete

Det är sedan länge känt att tekniker baserade på vektorrymdmodellen med en representation baserad på ord känd som *säck-med-ord* kan få klassificerare att uppnå state-of-the-art resultat. Men som namnet antyder ignorerar säck-med-ord modellen möjliga strukturer i texten eftersom den endast tar hänsyn till isolerade ord. Denna begränsning är välkänd och många försök har utförts för att bryta den genom att ta till mer avancerade metoder. Metoder som att inkludera detekteringen och indexeringen av egennamn, nominalfraser, fraser eller identifieringen av ordbetydelser. I dagsläget har de inte resulterat i några avgörande förbättringar [14].



Figur 1.1: En översikt över stegen som ingår i textklassificering.

I mitt arbete ska jag undersöka vilket bidrag egenskaper baserade på utdatan från syntaktiska och semantiska parsers – subjekt-verb-objekt (SVO)-tripplar och predikat-argument-strukturer – ger när de inkluderas i representationen. De här egenskaperna extraheras automatisk från texterna och kompletterar den ordbaserade representationen genom att lägga till semantiska och syntaktiska dimensioner.

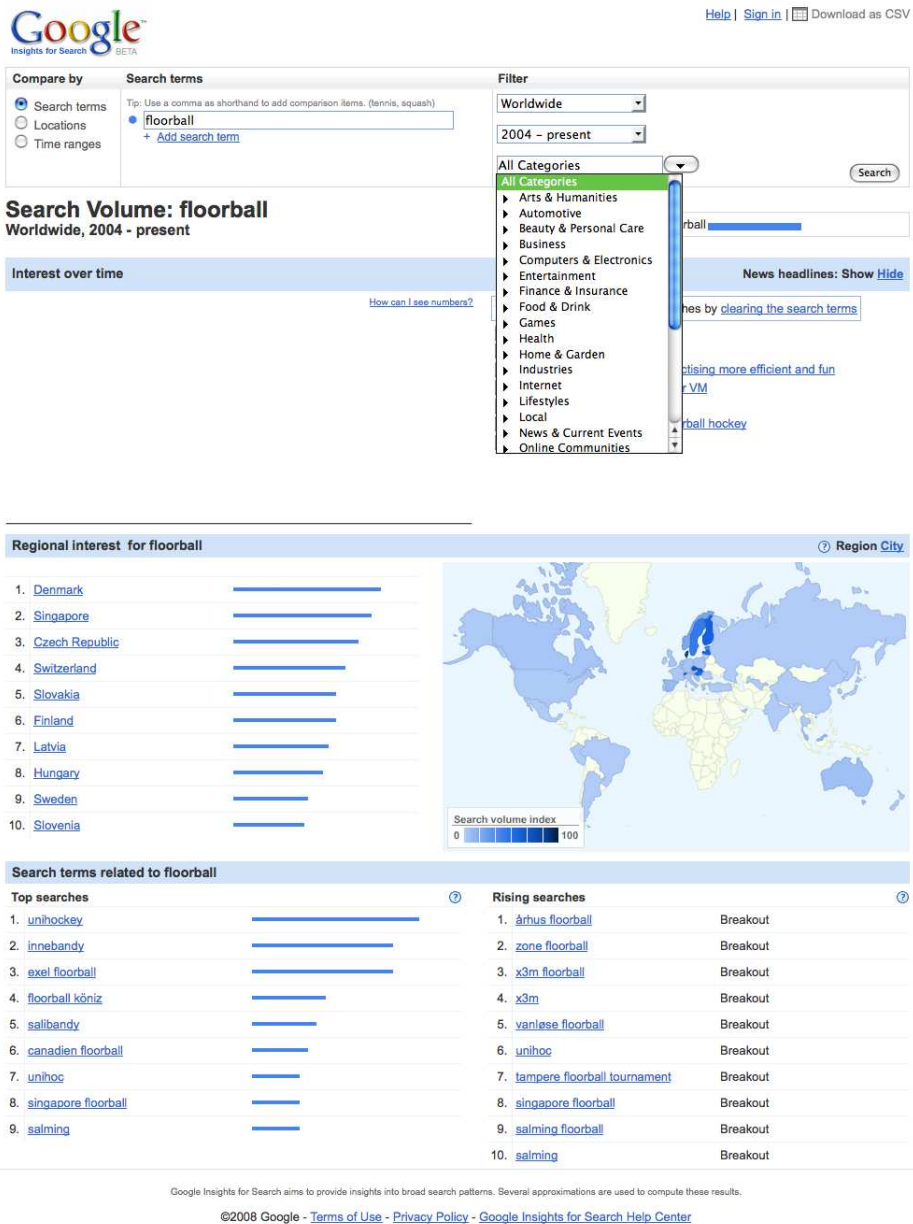
Jag kommer utföra mina experiment på Reuters korpus av nyhetsartiklar (RCV1) med ett standardiserat test [10].

1.3 Användningsområden

På internet finns enorma mängder information, den är till stor del ostrukturerad och utspridd vilket försvårar att den kommer till nytta. Textkategorisering kan i det här fallet strukturera upp informationen så att man lättare kan tillgodogöra sig den. Google har t.ex. en tjänst¹ som låter en göra sökningar där man kan filtrera sökresultaten baserat på vilka kategorier de tillhör, se Figur 1.2.

Ett annat användningsområde är automatiserad postsortering. E-post skulle t.ex. kunna kategoriseras och placeras i olika mappar. I större skala så kan ett företag skanna in sin fysiska post och även låta den bli automatisk sorterad. Andra tekniker kan även kombineras med textkategorisering för att t.ex. kunna läsa och svara på post utan några manuella steg.

¹<http://www.google.com/insights/search/>



Figur 1.2: Googles tjänst för att söka information med hjälp av kategorier.

Kapitel 2

Representationer av text för automatisk klassificering

2.1 Vektorrymdsmodellen

I statistisk klassificering används vanligtvis vektorrymdsmodellen för att representera text [19]. Den här modellen använder egenskaper extraherade från dokumentsamlingen för att bygga upp ett rum, där varje egenskap utgör en dimension. Varje enskilt dokument representeras av en vektor, där varje koordinat indikerar förekomsten av en specifik egenskap och även dess viktning. Dokumentvektorerna kan placeras i rummet och deras position kan användas för att bestämma vilken kategori dokumenten tillhör.

2.2 Ordbaserad representation

Att använda orden ur en text som egenskaper till vektorrymdsmodellen är den vanligaste representationen och kallas för *säck-med-ord*. Antag att vi har en dokumentsamling bestående av endast två dokument, vars innehåll är:

D1: Chrysler plans new investment in Latin America.

D2: Chrysler plans major investments in Mexico.

appliceringen av säck-med-ord modellen på samlingen använder alla orden som egenskaper och resulterar i dokumentvektorerna som kan ses i tabell 2.1. Orden har blivit reducerade till sina rötter och de vanligaste orden – stopporden – används inte, eftersom de oftast förekommer i alla dokument. För varje egenskap, anger vektorn hur många gånger den förekommer i dokumentet. Detta värdet kallas för *termfrekvens*, *tf*.

I Tabell 2.1, använde dokumentvektorerna den råa termfrekvensen för varje ord och gav därför alla ord lika stor betydelse. Men sällsynta egenskaper är oftast av mer betydelse än egenskaper som förekommer i många av dokumenten i samlingen. Spridningen av en egenskap mäts med dokumentfrekvensen, som definieras som antalet dokument där egenskapen förekommer. För att ge sällsynta egenskaper mer betydelse, viktas termfrekvensen med den *inverterade*

D#\ Words	chrysler	plan	new	major	investment	latin	america	mexico
1	1	1	1	0	1	1	1	0
2	1	1	0	1	1	0	0	1

Tabell 2.1: Dokumentvektorer baserade säck-med-ord modellen.

dokumentfrekvensen, idf (2.1). Detta viktningsformat kallas för $tf \times idf$ och det finns många varianter av den. För en lista av viktningsformat och en jämförande studie, se [18] och [8].

$$idf = \log \left(\frac{\text{collection size}}{\text{document frequency}} \right) \quad (2.1)$$

2.3 Utöka en representation med semantiska egenskaper

Ordbaserade representationer är enkla och robusta, men kommer också med begränsningar. Använder man en säck-med-ord representation ignorerar man fras och meningsorganisationen och deras logiska struktur. Intuitivt borde semantiken hos meningar i ett dokument vara till hjälp för att kategorisera det mer exakt. Jag extraherade semantiska egenskaper från varje korpusmening – predikat-argument-tupler, subjekt-verb-objekt-tripplar och ordbetydelse information – och utökade dokumentvektorerna med dem.

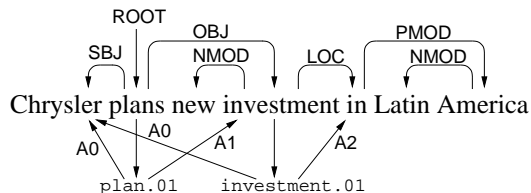
Predikat-argument-strukturer är basala konstruktioner i de flesta formalismer som sysslar med kunskapsrepresentation. Enkelt sett kan de ses som en aktion och komponenterna som är relevanta för aktionen. De är lika framträdande i lingvistiska teorier om kompositionell semantisk representation. I det enklaste fallet kan predikat-argument tuplar approximeras av subjekt-verb-objekt-tripplar eller subjekt-verbpar och extraheras från syntaktiska dependensträd.

Dependensträd är ett sätt att representera den syntaktiska strukturen i en mening. Ett dependensträd är uppbyggt av riktade bågar mellan orden i meningen som beskriver beroenden. I Figur 2.1 som visar ett dependensträd är *Chrysler* och *investment* dependent till *plan*. Varje ord har endast en ingående båge men kan ha flera utgående. Rotordet som oftast är huvud verbet i meningen har ingen ingående båge. Bågarna mellan orden i dependensstrukturer kan annoteras med grammatiska funktioner som subjekt och objekt.

SVO-representationer har använts i vektorrymd ansatser till ett antal problem [11, 15] inklusive text kategorisering [4] fast då på ett begränsat korpus av webbsidor. I det vida publicerade semantiska webbinitiativet, Berners-Lee et al. [1] förespråkade deras användande som ett *naturligt sätt att beskriva den stora majoriteten data behandlad av maskiner*.

Men syntaktiska parsträd och semantiska strukturer är generellt sätt inte isomorfiska och tuplar extraherade direkt från dependensträd är känsliga för parafrasering orsakad av lingvistiska processer. T.ex. en mening som skiftar från vara passiv eller aktiv, *Chrysler planned investments / investments were planned by Chrysler*. Eller skiftningar mellan objekt och indirekta objekt som i

dativ-skiftningar, *We sold him the car* / *We sold the car to him*.



Figur 2.1: Exempelmening med dependenssyntax och rollsemantisk annotering. Övre pilar visar dependensrelationer och de undre semantiska roller.

Rollsemantik [3] är en formalism som abstraherar den syntaktiska representationen genom att införa semantiska roller som AGENT och PATIENT istället för att använda grammatiska funktioner som subjekt och objekt.

Figure 2.1 visar den första exempelmeningen från Sek. 2.2 annoterad med syntaktiska beroenden och rollsemantisk information enligt Propbank [16] och Nombank [13] standarden. Verbet *plan* är ett predikat definierat i PropBanklexikonet, som har en lista över dess fyra möjliga kärnargument: A0, planner, A1, the thing planned, A2, grounds for planning, A3, beneficiary. På samma sätt är substantivet *investment* ett NomBank predikat vars tre kärnargument är: A0, investor, A1, theme och A2, purpose. Utöver kärnargumenten kan predikaten också tilldelas adverbial som platser eller tider.

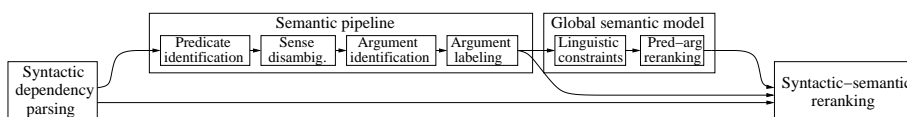
För varje predikat definerar PropBank och NomBank ett antal betydelser, som t.ex. *plan.01* och *investment.01* i exempelmeningen. Egenskaper baserade på betydelsen av ord, oftast WordNet-betydelser, har använts i textklassificering, men har inte resulterat i några avgörande förbättringar. För en genomgång av gångna studier och resultat, se [12].

Kapitel 3

Automatisk identifiering av semantiska roller

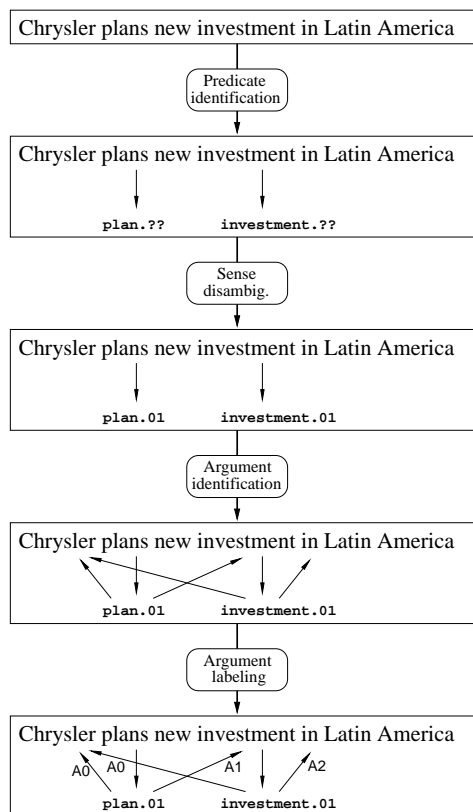
Rollsemantiska strukturer kan automatisk bli extraherade från fritext – uppgiften kallas för *semantic role labeling* (SRL). Trots att tidiga SRL-system [6] använde symboliska regler, förlitar sig moderna system på statistiska tekniker [5]. Detta har möjliggjorts av tillgängligheten av träningsdata, först från FrameNet [17] och senare PropBank och NomBank. Automatisk identifiering av semantiska roller kan nu appliceras på obegränsad text, åtminstone affärstexter, med tillfredsställande kvalitet.

Jag använde ett fritt tillgängligt SRL-system [9] för att extrahera predikat-argument-strukturerna¹. Systemet bygger på en syntaktisk och en semantisk subkomponent. Den syntaktiska modellen är en botten-upp dependensparser, och den semantiska modellen använder globala interferensmekanismer på en kedja av klassificerare. Den kompletta syntaktiska-semantiska utdatan väljs från en kandidatpool genererad av subsystemen. Figur 3.1 visar den övergripande strukturen och Figur 3.2 visar hur exempelmeningen bearbetas av den semantiska subkomponenten. Systemet fick den bästa poängen i den stängda utmaningen i CoNLL 2008 Shared Task [20]: syntaktisk märkning med exakthet på 89.32%, semantisk märkning med ett F_1 på 81.65 och en makro F_1 märkning på 85.49.



Figur 3.1: Arkitekturen över den semantiska rollmärkningssystemet.

¹Finns för nerladdning här: nlp.cs.lth.se



Figur 3.2: Exampel bearbetas av den semantiska kedjan.

Kapitel 4

Algoritmer för klassificering

Det finns en rad olika klassificerare som används inom textklassificering. Supportvektormaskiner(SVM) är de som har visats de bästa resultaten. Men i vissa tillämpningar är algoritmens snabbhet av yttersta vikt och då är även Rocchio och K-nn intressanta.

4.1 Supportvektormaskiner

Målet för en supportvektormaskin(SVM) är att separera positiva och negativa exempel som befinner sig i en vektorrymd med ett hyperplan. Ett vanligt problem i maskininlärning är att man anpassar sig för mycket mot träningsdatan, så kallad överträning. Man får då ett lågt antal felklassificerade exempel på träningsdatan men ett stort på data som man inte tränat på då man inte lyckats fånga de generella särskiljande egenskaperna. En SVM motverkar överträning genom att välja det separerande hyperplanet med störst marginal. Detta kan uttryckas som ett optimeringsproblem.

$$\begin{cases} \max \frac{k}{\|\vec{w}\|} \\ y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 \end{cases}$$
$$y_i = \begin{cases} +1 & \text{om } x_i \text{ är ett positivt exempel} \\ -1 & \text{om } x_i \text{ är ett negativt exempel} \end{cases}$$

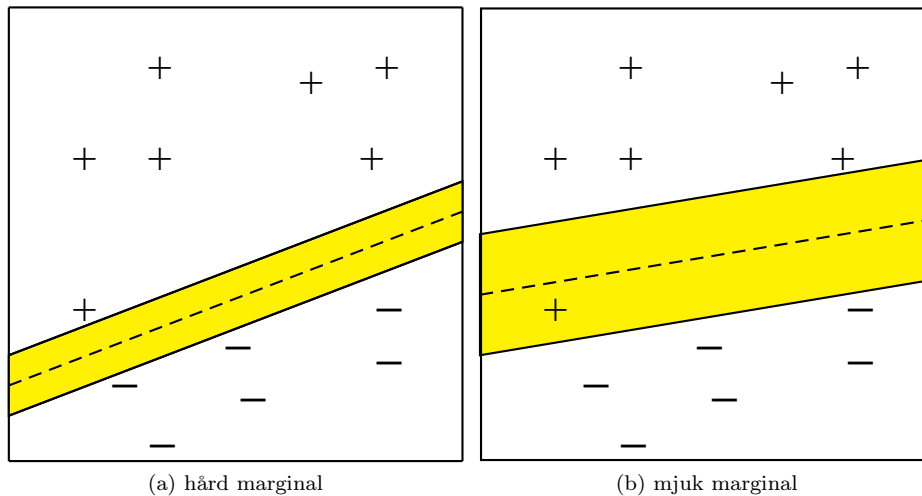
Hyperplanet definieras med ekvationen $\vec{w} \cdot \vec{x}_i + b = 0$ som har två parallella plan $\vec{w} \cdot \vec{x}_i + b = \pm k$ på varsin sida som definerar marginalen. Notera att $\vec{w}/\|\vec{w}\| \cdot \vec{x}_i$ är projektionen av punkten x_i på normalen till hyperplanet. Enligt definitionen av hyperplanet är $-b = \vec{w} \cdot \vec{x}_i$ för alla x_i som ligger på hyperplanet då följer att $-b/\|\vec{w}\| = \vec{w}/\|\vec{w}\| \cdot \vec{x}_i$ alltså är $b/\|\vec{w}\|$ avståndet från origo till hyperplanet.

Där \vec{w} och b beskriver hyperplanet och $\vec{w} \cdot \vec{x}_i + b$ är positivt eller negativt beroende på vilken sida av hyperplanet \vec{x}_i ligger.

Men data innehåller ofta brus, t.ex. felklassificerade exempel, detta kan hanteras genom att tillåta att vissa exempel befinner sig på fel sida av hyperplanet, en mjuk marginal så att säga. En sådan SVM har följande optimeringsproblem.

$$\begin{cases} \min \|\vec{w}\| + C \sum_{i=1}^m \xi_i^2 \\ y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 - \xi_i \\ \xi_i > 0, \quad i = 1, \dots, m \end{cases}$$

Här har man lagt till slack-variablerna ξ_i som tillåter att exempel befinns sig på innanför marginalen. Hur förlåtande man ska vara för felplacerade exempel kontrolleras av konstanten C . väljer man C till 0 får man en hård-marginal SVM.



Figur 4.1: största marginal.

4.2 K-nn

K-nearest neighbor (k-nn) är en mycket enkel algoritm som inte har något träningssteg, arbetet sker istället under själva klassificeringen. För att klassificera en dokumentvektor tittar man på de k närmaste dokumentvektorerna från träningsmängden och väljer den klassificeringen som är i majoritet.

4.3 Rocchio

Under träningssteget i Rocchio skapas det en prototypvektor för varje kategori. En prototypvektor är summan av alla vektorerna som tillhör en kategori. För att klassificera en dokumentvektor mäter man avståndet till prototypvektorerna och väljer kategorin som har den närmaste liggande prototypvektorn.

4.4 Naive Bayes

Naive Bayes är en sannolikhetsbaserad klassificerare som bygger på Bayes teorem. Man försöker uppskatta sannolikheten att texten tillhör kategorin C förutsatt värdena på egenskaperna F_1, \dots, F_n . Träningen sker genom att man utifrån

träningssmängden approximerar sannolikheterna för $p(F_i|C)$ och $p(C)$. Klassificeringen sker sedan genom välja den kategorin som har högst sannolikhet enligt Ekvation 4.1.

$$p(C|F_1, \dots, F_n) = p(C) \sum_{i=1}^n p(F_i|C) \quad (4.1)$$

Kapitel 5

Experimentuppställning

Jag utförde en serie av experiment för att avgöra bidragen från tre mängder av syntaktisk-semantiska egenskaper: ordbetydelse information, subjekt-verb-objekt-tripplar och rollsemantiska predikat-argument-tuplar. Alla testades genom att utöka den ordbaserade representationen i vektorrymd-modellen. Jag beskriver först datamängderna, sen experimentparametrarna och till sist siffrorna jag fick för olika kombinationer av egenskaper.

5.1 Korpus

Jag utförde mina experiment på RCV1-v2 [10] korpuset, som är en korrigerad version av RCV1 (Reuters Corpus Volume 1). Jag använde mig av LYRL2004-delningen, som placerar artiklar publicerade mellan augusti 20, 1996 till augusti 31, 1996 i träningsmängden och artiklar mellan september 1, 1996 till augusti 19, 1997 i testmängden. Jag utförde delningen på den ursprungliga RCV1-v1 samlingen vilket resulterade i 23,307 träningsdokument och 783,484 testdokument. RCV1 delar in kategorier i 3 huvudkategorier: regions, topics och industries. Regions innehåller geografiska platser som en artikel berör. Topics försöker fånga ämnena behandlade i en artikel medan industries anger de industriområden som nämns.

5.2 Klassificeringsmetod

Jag reproducerade förutsättningarna vid klassificeringsmetoden SVM.1 [10]. Jag använde SVM^{light} [7] klassificeraren med standard parameterarna och SCutFBR.1-algoritmen [21] för välja det optimala tröskelvärdet.

När SVM^{light} klassificerar dokumentvektorer tilldelar den dem ett värde mellan [-1,+1]. För att klassificera vektorerna kan man sedan märka alla vektorer vars värde är positivt som medlemmar i kategorin. Men att använda 0 som tröskelvärdet ger oftast inte det bästa resultatet, därför använder jag SCutFBR.1-algoritmen för att hitta ett bra tröskelvärde.

Med SCut tränar man klassificeraren på en del av träningsmängden och klassificerar den resterande mängden. Eftersom man vet de korrekta kategorierna för exemplen i träningsmängden kan man för varje kategori hitta det tröskelvärdet mellan [-1,+1] som ger den största andelen rätta klassificeringar (se Figur 5.1).

Tröskelvärdena man hittar här använder man sen för att klassificera testmängden. För att öka chanserna till bra tröskelvärden som inte beror på uppdelningen man gjorde av träningsmängden utförde jag 5-delad korsvalidering. Då får jag fem tröskelvärden per kategori, jag använder medelvärdet av dem.

SCutFBR.1 är en utökning av SCut för hantera kategorier som har få eller dåliga träningsexempel. Det farliga fallet man vill undvika är när en kategori tilldelas ett för lågt tröskelvärde. När en kategori felaktigt tilldelas ett lågt tröskelvärde är risken stor för många falskalarm när testmängden klassificeras, dvs. kategorin tilldelas felaktigt till många av dokumenten. För att undvika detta väljer man ett fbr-värde och om F_1 värdet är lägre än detta för en kategori väljer man det högsta värdet som tilldelades något av exemplen som tröskelvärde.

+	-	+	+	+	-	-	-
0.2	0.23	0.33	0.34	0.39	0.45	0.57	0.62

Figur 5.1: Det optimala tröskelvärdet är 0.39 eftersom då klassificeras det maximala antalet dokument rätt.

Valet av ett fbr-värde gjordes genom att utföra SCutFBR.1 algoritmen för varje värde på fbr inom $[0, 0.8]$ med 0.1 steg. Värdet som gav det högsta F_1 mikro/makromedelvärdet på träningsmängden valdes. Även här gjordes testerna med 5-delad korsvalidering och det var det medelvärdena som jämfördes.

5.3 Korpustagging och parsers

Jag annoterade RCV1-korpusen med POS-taggar, dependensrelationer och predikatargument-strukturer med SRL-systemet som nämndes i Sek. 3. Pos-taggar använde tekniker liknande dem beskrivna av Collins [2].

5.4 Egenskapsmängder

Jag utförde mina experiment med tre grupper av egenskapsmängder. Den första egenskapsmängden är baslinjens säck-med-ord. Den andra innehåller tripplar bestående av subjekt, verb och objekt (SVO) för givna predikat. Den tredje mängden utgörs av predikat, deras betydelse och deras mest frekventa argument: A0 och A1. Jag demonstrerar egenskaperna med meningen *Chrysler plans new investment in Latin America*, vars syntaktiska och semantiska-graf visas i Figur 2.1.

Som första egenskapsmängd använde jag säck-med-ord som kom från den pre-tokeniserade versionen av RCV1-v2, släppt tillsammans med [10], utan någon ytterligare modifiering. Exempel av säck-med-ord egenskaper visas i Tabell 2.1.

Till den andra egenskapsmängden, SVO-triplarna, övervägde jag verben med Penn-TreeBank taggarna: VB, VBD, VBG, VBN, VBP och VBZ. I varje mening i korpusen och för varje verb extraherade jag deras subjekt och objekt-huvuden från dependensparser utdatan. De här dependenterna kan ha andra grammatiska funktioner knutna till sig. Jag valde subjekt och objekt eftersom de

oftast motsvarar de primära semantiska rollerna. Jag skapade egenskapsymbolerna genom att konkatenera varje verb med dess subjekt och objekt beroenden när de existerar. Verb utan något objekt- eller subjektberoenden ignoreras. Egenskapen skapad från exempelmeningen är: *plan#Chrysler#investment*.

Den tredje egenskapsmängden innehåller predikaten från korpuset och deras mest frekventa argument. Jag använde den semantiska utdatan från SRL-systemet för att identifiera alla verb och substantiv beskrivna i PropBank och NomBank-databaserna samt deras 0 och 1 argument. Jag kombinerade dem för att skapa fyra olika delmängder av semantiska egenskaper. Den första delmängden innehåller endast predikatbetydelsena. Jag skapade dem genom att suffixa predikatorden med deras betydelenummer som t.ex. *plan.01*. De tre andra delmängderna utgörs av predikaten samt ett eller två av deras primära argument, *argument 0* och *argument 1*. Som vid VSO-tripplarna skapade jag egenskapsymbolerna genom konkatenering av predikaten och argumenten. De tre olika kombinationerna jag använde är:

1. Predikatet och dess första argument, argument 0. I exemplet, *plan.01#Chrysler*
2. Predikatet och dess andra argument, argument 1. I exemplet, *plan.01#investment*
3. Predikatet och dess första och andra argument, argument 0 och 1. I exemplet, *plan.01#Chrysler#investment*

Alla representationer skapade ur de här egenskapsmängderna var viktade med viktningsschemat $\log(tf) \times idf$. Dessutom ersattes bestämda pronomen, nummer och årtal av symboler för respektive klass.

Kapitel 6

Resultat

6.1 Evalueringsmetod

Eftersom artiklarna i RCV1 kan vara märkta med flera olika kategorier utförde jag en multilabel-klassificering. En sådan utförs genom att applicera en klassificerare på varje kategori och sen sammanställa resultaten från dem. För klassificeringen av en enskild kategori i , kan resultaten representeras av en kontingenstabell (Tabell 6.1) och från den här tabellen kan vi beräkna standardmåten *Precision* och *Recall*. Jag sammanfattade resultaten med det harmoniska medelvärdet F_1 av *Precision* och *Recall* (6.1).

	+ exempel	- exempel
+ kategoriserad	a_i	b_i
- kategoriserad	c_i	d_i

Tabell 6.1: Resultaten av en kategorisering representerad i en kontingenstabell.

$$\begin{aligned} Precision &= \frac{a_i}{a_i + b_i} \\ Recall &= \frac{a_i}{a_i + c_i} \end{aligned} \tag{6.1}$$

Precision ger ett värde på hur stor andel av exemplen man kategoriserade som positiva verkligen är positiva. *Recall* ger ett värde för hur många av de positiva exemplen man lyckades identifiera. Men när man vill jämföra resultat är det en fördel om man kan uttrycka resultat med ett enda värde därav det harmoniska medelvärdet F_1 .

För att mäta resultatet över alla kategorierna, använde jag F_1 mikromedelvärde och F_1 makromedelvärde. Makromedelvärdet beräknas genom att ta medelvärdet av F_1 över alla kategorierna (6.4), medan mikromedelvärdet beräknas genom att summera ihop alla de binära valen (6.2) och beräkna F_1 från det (6.3).

$$\mu Precision = \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n a_i + b_i} \quad (6.2)$$

$$\mu Recall = \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n a_i + c_i}$$

$$\mu F_1 = \frac{2 \times \mu Precision \times \mu Recall}{\mu Precision + \mu Recall} \quad (6.3)$$

$$maF_1 = \frac{1}{n} \sum_{i=1}^n F_1^i \quad (6.4)$$

6.2 Resultat

De sex egenskapsmängderna möjliggör 64 olika representationer av min data. Jag tilldelade varje representation en kod med en sex tecken lång sträng där en 1 på den första position indikerar att säck-med-ord mängden är inkluderad osv. enligt Tabell 6.2

Egenskapsmängd	Kod
Bag of words	100000
Predicates	010000
VSO triples	001000
Argument 0	000100
Argument 1	000010
Arguments 0 and 1	000001

Tabell 6.2: Koder för egenskapsmängderna. En kod för en representation är resultatet av en bitvis och-operation mellan koderna av de inkluderade egenskapsmängderna.

För att få en approximation av prestandan av representationerna utförde jag tester på träningsmängden. Jag utförde sedan det fullständiga testet på de representationer som visade de mest lovande resultaten. Jag mätte och optimerade för F_1 mikro och makromedelvärde. Tabellerna i figur 6.1 visar representationerna jag valde ut från de inledande testerna och deras resultat i det fullständiga testet. Representationerna som inkluderade säck-med-ord, predikat och en eller flera av argumentmängderna eller VSO-mängden åstadkom de bästa resultaten.

Effektiviteten av de enskilda kategorierna visas i Figurerna 6.2 och 6.3. Kategorierna är sorterade efter storleken på träningsmängden. Graferna har slätats ut med en lokal linjär regression inom ett område på $[-200, +200]$.

Eftersom figurerna ligger tätt ihop i Figur 6.2 och 6.3 visar jag den relativa felminskningen i Figur 6.4 och 6.5.

Jag utförde McNemar-testet för att mäta signifikansen av felminskningen. I Figur 6.6 visar tabellerna hur många kategorier som hade en signifikans under 0.95 av totalt 103 kategorier.

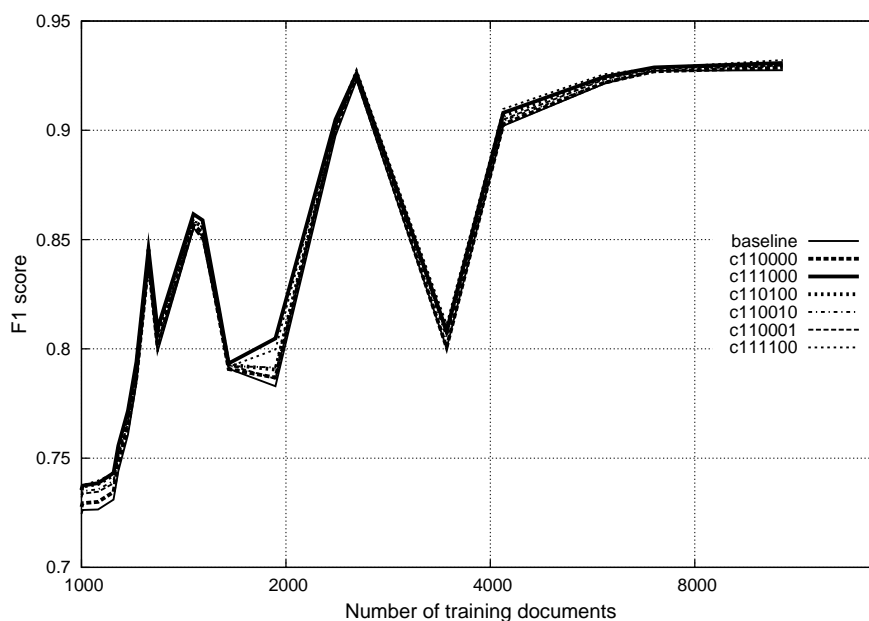
Egenskapsmängd	F_1
Baseline	81.76
c110000	81.99
c111000	82.27
c110100	82.12
c110010	82.16
c110001	81.81
c111100	82.17

(a) Resultat för F_1 mikromedelvärde

Egenskapsmängd	F_1
Baslinje	62.31
c110000	62.09
c111000	62.57
c110100	62.16
c110010	62.77
c110001	62.24
c111100	62.44

(b) Resultat för F_1 makromedelvärde

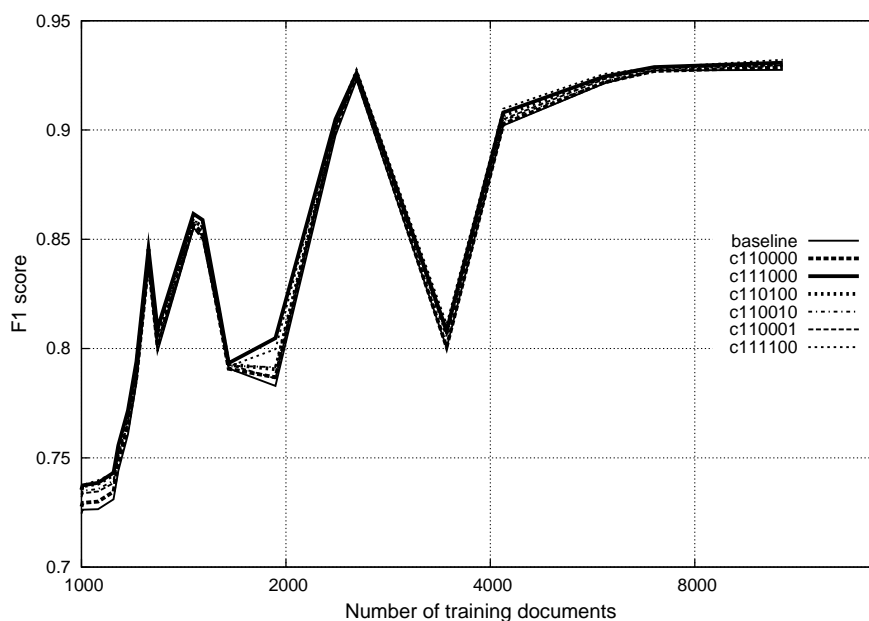
Figur 6.1: Effektiviteten av F_1 makro och mikromedelvärde på de mest lovande representationerna. Parametrarna sattes till att optimera F_1 mikro resp. makromedelvärdet. Baslinjens siffror motsvaras av en säck-med-ord.



Figur 6.2: F_1 värdet per kategori på mikromedelvärde optimerade klassificeringar. Grafen visar kategorier med fler än 1000 exempel i träningsmängden.

6.3 Slutsats

Jag har visat att komplexa semantiska egenskaper kan användas för att åstadkomma betydande förbättringar i textklassificering över en baslinje med säck-med-ord representation. De bästa resultaten i genomsnitt fås genom att utöka vektorrymd-modellen med dimensioner som representerar disambiguerade verb-predikat och SVO-tripplar. För kategorier som har mer än 2500 träningsexempel, ger tillägget av argument 0 de bästa resultaten. Kategorin som fick den största felminskningen var C15 och dess underkategorier speciellt C152.



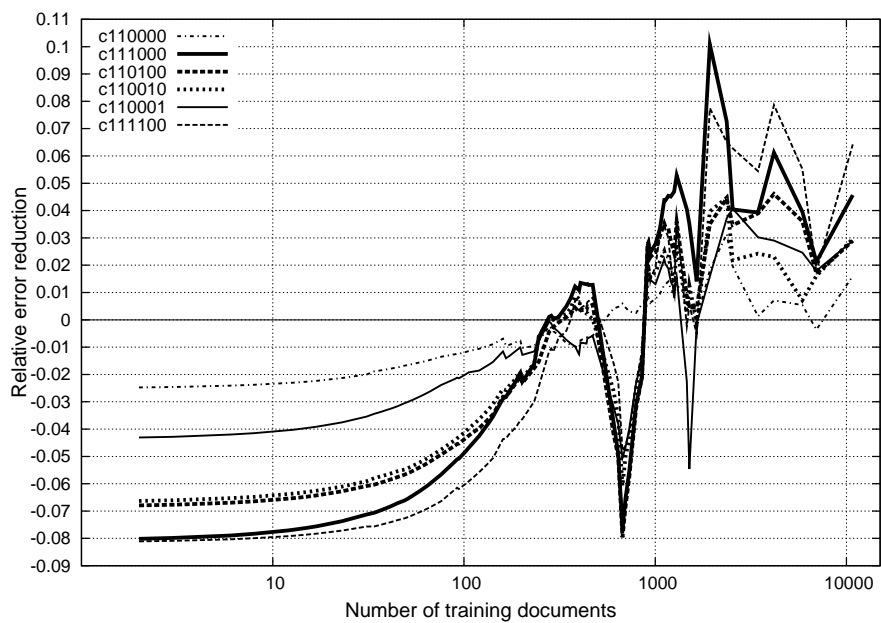
Figur 6.3: F_1 värdet per kategori på makromedelvärde optimerade klassificeringar. Grafen visar kategorier med fler än 1000 exempel i träningsmängden.

I experimenten utförda av Lewis et al., vars experimentuppställning jag kopierade, uppnåddes 81.6 som det högsta F_1 mikromedelvärde för topics kategorierna. Min baslinje som är en kopia av uppställningen som nådde bäst resultat fick 81.76 i samma test vilket kan bero på att jag använde en nyare version av SVM^{light} .

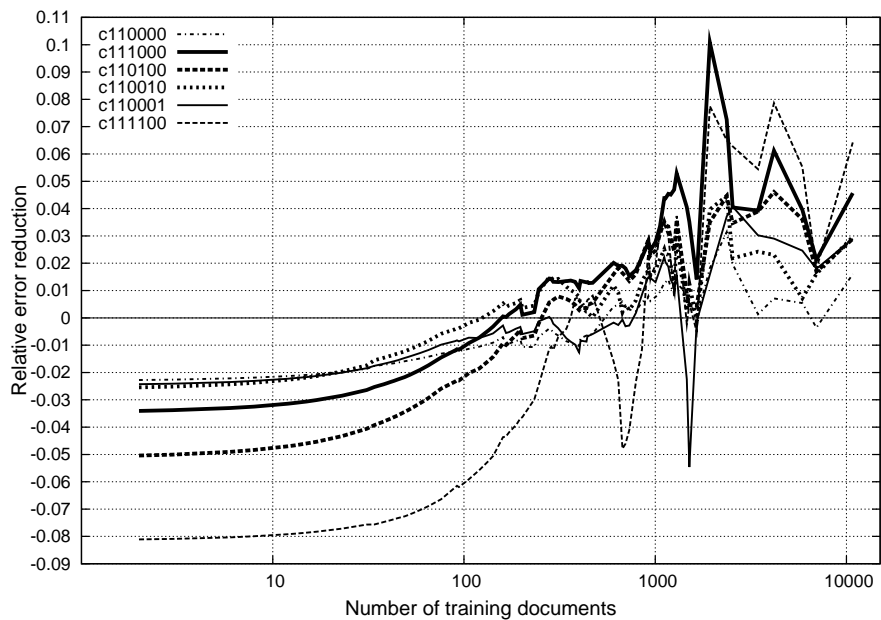
I motsats till tidigare studier [12], ger disambigueringen av predikatens betydelse en förbättring över baslinjen. En möjlig förklaring kan vara att:

- PropBank/NomBank databaserna har enklare uppdelning av betydelserna än WordNet, t.ex. *plan* har fyra olika betydelser i WordNet och endast en i PropBank; *investment* har sex olika betydelser i WordNet men bara en i NomBank.
- Penn TreeBank korpuset på vilket den semantiska parsern tränas är större än SemCor, korpuset som vanligtvis används för att träna ordbetydelse-disambigueringssystem. Detta betyder att klassificeraren jag använde kan vara mer precis.

Tänkbara fortsättningar på mitt arbete är att: undersöka vilken effekt olika viktningsscheman har på de grammatiska och syntaktiska egenskaper jag undersökt, experimentera med predikat-argument-egenskaper som bygger på andra argument än 0 och 1 samt att göra experiment på fler korpus.



Figur 6.4: Den relativa felminskningen per kategori för mikromedelvärde optimerade klassificeringar.



Figur 6.5: den relativa felminskningen per kategori för makromedelvärde optimerade klassificeringar.

Egenskapsmängd	< 0.95
c110000	27
c111000	23
c110100	23
c110010	26
c110001	25
c111100	25

(a) signifikans för F_1 mikromedelvärde

Egenskapsmängd	< 0.95
c110000	23
c111000	20
c110100	21
c110010	25
c110001	25
c111100	22

(b) signifikans för F_1 makromedelvärde

Figur 6.6: Antalet kategorier av totalt 103 som hade en signifikans under 0.95 när parametrarna var satta till att optimera F_1 mikro resp. makromedelvärde.

Bilaga A

Ordlista

- **predikat** - är vanligtvis verbet i en mening men i meningar som *Karl-Gustav är kungen i Sverige* är substantivet *kungen* predikatet. Ett exempel där predikatet är markerat, *Kalle **sparkade** bollen*.
- **subjekt** - i meningar kan oftast lokaliseras genom att ställa sig frågan om vem som utför handlingen som beskrivs av predikatet. Ett exempel där subjektet är markerat, ***Kalle** sparkade bollen*.
- **objekt** - anger någon eller något som ingår i subjektets handling av verbet. Ett exempel där objektet är markerat, *Kalle sparkade **bollen***.
- **nominalfras** - är en fras som fungerar som ett substantiv. Exempel på nominalfraser är *boll* och *den runda bollen*.
- **egennamn** - är namn på personer, städer, företag etc.
- **parafrasering** - omskrivning av en text.
- **bestämda pronomen** - pekar ut något i en text. Exempel på bestämda pronomen är *hon*, *jag*, *de* och *våran*.
- **adverbial** - är en satsdel som beskriver tid, plats, sätt, etc. Adverbial delas ofta upp i olika klasser baserat på vad de beskriver som tidadverbial, rumsadverbial, orsaksadverbial, etc.
- **dativ** - anger att ett ord en i en mening fungerar som ett indirekt objekt vilket både i engelska och svenska ofta markeras med orden *till*, *åt* eller *för*.

Litteraturförteckning

- [1] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, pages 29–37, May 2001.
- [2] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 2002.
- [3] Charles J. Fillmore. The case for case. In E. R. Bach and R. T. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston, 1968.
- [4] Johannes Fuernkranz, Tom Mitchell, and Ellen Riloff. A case study in using linguistic phrases for text categorization of the www. In *In Working Notes of the AAAI/ICML Workshop on Learning for Text Categorization*, 1998.
- [5] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.
- [6] Graeme Hirst. A foundation for semantic interpretation. In *Proceedings of the ACL-1983*, 1983.
- [7] Thorsten Joachims. Making large-scale SVM learning practical. In Bernhard Schölkopf, Christopher Burges, and Alexander Smola, editors, *Advances in Kernel Methods. Support Vector Learning*. MIT Press, 1999.
- [8] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, Boston, 2002.
- [9] Richard Johansson and Pierre Nugues. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings of CoNLL–2008*, 2008.
- [10] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
- [11] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL*, 1998.
- [12] Trevor Mansuy and Robert J. Hilderman. A characterization of wordnet features in boolean models for text classification. In *Proceedings of the Fifth Australasian Data Mining Conference (AusDM2006)*, pages 103–109, 2006.

- [13] Adam Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. The nombank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.
- [14] Alessandro Moschitti and Roberto Basili. Complex linguistic features for text classification: a comprehensive study. In Sharon McDonald and John Tait, editors, *Proceedings of ECIR-04, 26th European Conference on Information Retrieval*, Sunderland, 2004.
- [15] Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–299, 2007.
- [16] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: an annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, 2005.
- [17] Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. *Framenet II: Theory and practice*. <http://framenet.icsi.berkeley.edu/book/book.html>. Printed June 20, 2006, 2006.
- [18] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical Report TR87-881, Department of Computer Science, Cornell University, Ithaca, New York, 1987.
- [19] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Technical Report TR74-218, Department of Computer Science, Cornell University, Ithaca, New York, 1974.
- [20] Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The CoNLL–2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL–2008*, 2008.
- [21] Yiming Yang. A study on thresholding strategies for text categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 137–145, New Orleans, 2001. ACM Press.