

Institutionen för lingvistik och filologi  
Språkteknologiprogrammet  
Examensarbete i datorlingvistik

5 april 2005

# Representation av text med semantiska vektorrumsmodeller

Marcus Ericsson

Handledare  
Jussi Karlgren, SICS  
Magnus Sahlgren, SICS  
Beáta Megysi, Uppsala Universitet

## Sammandrag

För att representera text kan man använda sig av semantiska vektorrumsmodeller. Dessa modeller använder sig av information om ordförekomst i en text för att datamaskinellt skapa denna representation av textens språk.

Denna magisteruppsats kommer att introducera läsaren till vektorrymssemantik samt ett flertal vektorrymssemantiska metoder för att skapa semantiska representationer datamaskinellt. Syftet med detta arbete är att på uppdrag av företaget INNOVEAS skapa en semantisk representation av en teknisk textsamling. För att åstadkomma detta introduceras en ny vektorrymsrepresentationsmodell, *unär indexing*, som är skapad för att med fördel kunna behandla mindre datamängder.

Arbetet visar att trots en god representationsmodell är den textuella kvalitén på måltexterna avgörande för att modellen ska prestera goda resultat.

## Tack

Detta arbete har finansierats av SICS, Swedish Institute for Computer Science, och utförts på Institutionen för lingvistik och fliologi vid Uppsala Universitet. Jag vill framföra mitt tack till mina båda handledare på SICS, Jussi Karlgren och Mangnus Sahlgren, utan deras stöd och instruktion skulle detta arbete inte vara det som det nu är. Ett stort tack även till min akademiska handledare Beáta Megyesi för hennes hjälp med uppsatsen i dess slutgiltiga fas. Jag skulle även vilja tacka de som med sitt stöd och hjälp underlättade detta arbetes fullbordning. Bland dessa märks Filip von Kartaschew, Magnus Boman och Björn Gambäck.

Slutligen vill jag tacka Emma Eriksson ♡.

# Innehåll

<b>1</b>	<b>Introduktion</b>	<b>4</b>
1.1	Syfte . . . . .	4
1.2	Disposition . . . . .	4
<b>2</b>	<b>Vektorsemantiska modeller</b>	<b>6</b>
2.1	Vektorsemantisk metod . . . . .	6
2.2	Ordrepresentation . . . . .	7
2.3	Distributionshypotesen . . . . .	7
2.4	Mening . . . . .	8
2.5	Begreppsrepresentation . . . . .	9
2.5.1	Från ord till ord . . . . .	10
2.5.2	Metoder för begreppsrepresentation . . . . .	10
2.6	Kontextuella modeller . . . . .	13
2.6.1	Fönsterträning . . . . .	13
2.6.2	Dokumentträning . . . . .	13
2.7	Unär indexering . . . . .	14
2.8	Analysverktyg . . . . .	14
<b>3</b>	<b>Experiment</b>	<b>15</b>
3.1	Data . . . . .	15
3.2	Förprocessering . . . . .	15
3.2.1	Normalisering . . . . .	15
3.3	Experimentutförande . . . . .	16
3.3.1	Experimentbelysning med demonstrationsdokument . . . . .	17
3.3.2	Delexperiment . . . . .	19
3.4	Utvärdering . . . . .	19
3.4.1	Vektorlikhet . . . . .	20
3.4.2	Utvärderingsmetod . . . . .	21
<b>4</b>	<b>Resultat</b>	<b>23</b>
<b>5</b>	<b>Diskussion</b>	<b>24</b>
<b>6</b>	<b>Sammanfattning</b>	<b>26</b>

## Tabeller

1	Ett litet semantiskt rum . . . . .	18
2	Ordrepresentationens resultat . . . . .	23
3	Begreppsrepresentationens resultat . . . . .	23

## Figurer

1	Modell över representationsgenerering . . . . .	6
2	Exempel på fönsterträning . . . . .	13
3	Ett dokument före normalisering . . . . .	15
4	Ett dokument efter normalisering . . . . .	16
5	Ett demonstrationsdokument . . . . .	17
6	Demonstrationsdokumentets dokumentvektor . . . . .	19
7	Ett enkelt vektorrum före och efter normalisering . . . . .	21
8	Kosinusvärde och Euklidiskt avstånd mellan två vektorer . . . . .	21

# 1 Introduktion

En semantisk representation modellerar språk från text och lagrar i dessa modeller information om hur ord, eller dokument, förhåller sig till varandra i texten eller till texten själv. Dessa representationer kan användas i flertalet olika användningsområden där det är centralt att kunna utvärdera ny textuell information mot redan existerande information, som i till exempel textklassificering.

Repräsentationsgenerering av text är processen att skapa en semantisk representation av en text som modellerar denna texts språk. Under det senaste decenniet har vektorrymsmodeller börjat spela en allt större roll inom detta fält. Huvudidén är att representera varje ord som en vektor vars förekomster i texten utgör vektorns särdrag. Att representera ord som vektorer i ett flerdimensionellt teoretiskt rum har fördel av att det lätt går att mäta dessa ords likhet till varandra med hjälp av matematiska avståndsmått.

Det som ligger till grund för detta arbete är ett uppdrag från det tyska företaget INNOVEAS. Detta företag har behov av att skapa ett datamaskinellt hjälpmedel för att användas vid textklassificering. Grundtanken är att en mänsklig klassificerare skall kunna få förslag från ett analysverktyg och på så sätt kunna underlätta sitt arbete. Detta verktyg skulle kunna minska tiden det tar en mänsklig klassificerare att arbeta igenom och tillsätta klasser (även kallade nyckelord) till ett dokument. För att undersöka förutsättningarna till ett sådant verktyg har företaget SICS skapat detta examensarbete.

Denna uppsats kommer att fokusera på att tillverka en representation av en text bestående av en mängd små dokument. Denna representation skapas för att undersöka möjligheterna till att skapa ovan nämnda verktyg. Representationen kommer att skapas med hjälp av två vektorsemantiska metoder. Dessa metoder är den mer traditionella s.k. ordrepresentationen samt en nyare metod, begreppsrepresentation, som vid flera tillfällen visat stor potential i modern forskning (se exempelvis Karlgren & Sahlgren (2001), Landauer & Dumais (1997), Burgess, Lund & Atchley (1995)). För detta ändamål skapas en ny metod, *unär indexering* som är baserad på befintliga metoder men som är mer lämpad till att analysera mindre datamängder.

## 1.1 Syfte

Arbetets syfte är att skapa en representation av en textsamling bestående av knappt 3000 dokument. Representationen ska skapa en modell över ords likhet jämfört med andra ord baserat på dessa ords ordfrekvens. Denna representation kommer att utvärderas med avseende på dess överensstämmelse mot textens redan fördefinierade klassificering. Denna utvärdering kommer att ske med hjälp av en form av nyckelordsextraktion av framträdande termer ur den skapade representationen. Vidare kommer den nya metodens resultat att utvärderas mot den mer etablerade och klassiska ordrepräsentationsmetoden.

## 1.2 Disposition

I denna uppsats kommer efter denna introduktion att i kapitel 2 först att göras en genomgång av den teoretiska bakgrunden till vektorbaserad semantisk analys samt en genomgång av den så kallade *distributionshypotesen* som är den

bakomliggande teorin till den begreppsrepresentation som denna uppsats bygger kring. Kapitlet ger även en kort sammanfattning av tre uppmärksammade andra metoder för begreppsrepresentation. Sedan följer en genomgång av den vektorsemanstiska metod som kommer att nyttjas i de följande experimenten. Kapitel 3 innehåller en beskrivning av datan samt den preprocessoring och normalisering som utfördes på den. Vidare redogörs för de genomförda experimenten samt den utvärderingsmetod som kommer att användas. Detta kapitel följs av kapitel 4 där de experimentella resultaten redovisas. Uppsatsen avslutas med en diskussion, kapitel 5 samt slutligen med en sammanfattning av arbetet, kapitel 6.

## 2 Vektorsemantiska modeller

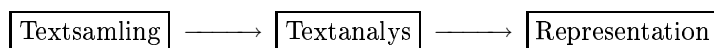
Det primära syftet med denna uppsats är att för företaget INNOVEAS skapa en representation av en textsamling med avseende på semantisk likhet. Denna representation kommer att skapas genom en vektorsemantisk modell där relativ ordfrekvens för ord jämförs mot varandra. I detta kapitel beskrivs den vektorsemantiska metodiken samt innehåller även en diskussion kring vilka berörningspunkter denna metodik har till den språkfilosofiska uppfattningen av mening. Vidare kommer en genomgång av ett antal kända metoder för att skapa en sådan representation att göras och slutligen kommer den metod som denna uppsats bygger på presenteras.

### 2.1 Vektorsemantisk metod

Vektorrymmodeller samlar information om de texter som de är baserade på i matriser bestående av förekomstinformation. Dessa matriser kan läsas som vektorer i en flerdimensionellt rum vars dimensionalitet utgörs av den aktuella textsamlingens vokabulär alternativt antalet dokument i textsamlingen.

I figur 1 nedan kan man se ett förenklat diagram över hur en representation skapas. En textsamling genomgår en *textanalys* med avseende på termfrekvens, antingen i avseende på textsamlingens dokument eller relativt mot alla andra termer i textsamlingen. Denna fas inleds med termselektion. Det är i denna fas som man avgör vilka element i textsamlingen som skall utgöra termen, det analyserade elementet i textsamlingen. Dessa termer kan bestå av ett antal olika element, allt ifrån hela fraser ner till stammar eller trunkerade graford. Det finns för och nackdelar med varje val av term, de större termerna (som flerordsfraser eller  $n$ -gram) har goda semantiska kvalitéer men dåliga statistiska egenskaper. De små termerna lider av det omvända förhållandet. I de kommande experimenten kommer graford att användas som term eftersom de kombinerar de statistiska och semantiska egenskaperna i något av en medelväg.

Efter denna analys blir varje term i textsamlingen associerad med en vektor som beskriver dess betydelse med avseende på textsamlingens dokument eller textsamlingens samtliga andra termer. En vektor är en geometrisk entitet som innehar en storlek (kallas även magnitud) samt en riktning i rummet. Dessa vektorer samlas i en matris som utgör en *representation* av den textsamling man valt att analysera. I denna representation (eller språkmodell) är varje term beskriven som en vektor av särdrag. Dessa särdrag kan till exempel utgöras av förekomststatistik.



Figur 1: En modell över representationsgenerering

Då varje term i den textmängd man valt att modellera tilldelas en vektor kan man genom att applicera ett avståndsmått i rummet matematiskt bestämma hur lika dessa termer är andra termer i vektorrummet. Eftersom dessa vektorer är definierade av hur de termer de är associerade med förekommer, får således vektorrummet helt olika utseende beroende på vilka träningsdata som använts.

I denna uppsats huvudexperiment kommer en textmängds termer att tilldelas två olika vektorer, en kontextvektor samt en indexvektor. Kontextvektorn är den vektor som efter träningsfasen representerar ordets semantiska egenskaper. Denna vektor byggs upp genom att addera in närliggande ord indexvektorers värde. Indexvektorerna är unika<sup>1</sup> för varje ord och tilldelas i och med *textanalysen*. Detta kommer att förklaras vidare i avsnitt 2.5.2.

## 2.2 Ordrepresentation

Den klassiska metoden för att skapa textuella representationer av text är en metod som fokuserar på termer. Detta innebär att varje term i texten får någon form av distributionell representation som speglar dess förekomst i texten. Dessa termer kan vara ett antal olika lingvistiska företeelser som till exempel stammar, lemma, graford,  $n$ -gram eller fraser. Av dessa metoder är en av de enklaste den s.k. *bag-of-words* metoden där en term representeras av en vektor med antalet undersökta dokument som kardinalitet<sup>2</sup>. Denna metod använder oftast graford eller ordstammar som term vilka oftast viktas med ett termviktningsschema baserade på  $tf \times idf$ .  $tf \times idf$  är ett väl utbrett standardmått som ofta används för informationsåtkomst där  $tf$  representerar *termfrekvensen* och  $idf$  är *inversen av dokumentfrekvensen*. (Sebastini 2002)

I denna typ av representation representeras en term av en vektor  $\vec{t}$  bestående av textens viktade termer på formen  $\vec{t}_i = (w_1 \dots w_n)$ , där  $w_n$  representerar en termvikt  $w$  för termen med avseende på dokumentet  $n$ . Oftast representeras dessa vikter,  $w_n$ , av ett  $tf \times idf$  värde av orden i texten.

Detta  $idf$  värde beräknas med formeln  $\frac{N}{n_i}$  där  $N$  är det totala antalet dokument och  $n_i$  är antalet dokument som det aktuella ordet  $i$  förekommer i. De kommande experimenten kommer att använda en av de mest förekommande varianterna av  $tf \times idf$  formeln som formuleras  $w_i = tf_i \times \log \frac{N}{n_i}$  (Robertson & Sparck Jones 1996).

Det är viktigt att notera här att en terms  $tf \times idf$  är en representation som helt bortser från textens strukturella information, som till exempel ordföljd eller annan kontextuell information. Denna metod ser endast till, och analyserar, ordens ytform. Det har utförts ett antal experiment med att förbättra den klassiska ordrepresentationen med hjälp av sådana metoder som till exempel:  $n$ -gram eller fraser i stället för ord som fokalelement eller försök med synonymkluster. Dessa försök har dock inte lyckats producera något avsevärt bättre resultat än den klassiska representationen. Utöver detta är dessa augmentationer av  $tf \times idf$  typiskt mer datamaskinellt kostsamma än den klassiska metoden (Sahlgren & Cöster 2004).

## 2.3 Distributionshypotesen

Hypotesen om att modellera betydelse genom distributionell information (härdaneftter kallad distributionshypotesen) är formulerad av matematikern och lingvisten Zellig Harris. Hypotesen är en modell för språklig betydelse genom analys

---

<sup>1</sup>I praktiken kommer samtliga eller nästan samtliga kontextvektorer att vara unika i jämförelse med varandra men till skillnad från indexvektorerna har dessa en teoretisk möjlighet att vara lika, om de förekommer i exakt lika kontexter.

<sup>2</sup>Med kardinalitet menas det antal särdrag som utgör vektoren

av den distributionella information som finns i språket i form av samförekomststatistik. Han uttrycker denna hypotes i boken *Mathematical Structure of Language*:

”...the meaning of entities, and the meaning of grammatical relations among them, is related to the restriction on combinations of these entities to other entities.” (Harris 1968, sid. 12)

Harris menar att om ett språkligt element, till exempel ordet A förekommer i samma eller i liknande distributioner som ordet B men inte ordet C, kommer dessa ords mening att korrelera med denna grad av liknande distribution. Med andra ord kan man genom att matematiskt uppskatta denna skillnad av distributionell förekomst, enligt hypotesen, uttala sig om dessa ords grad av likhet till varandra.

Harris väljer att uttrycka detta på följande sätt:

”If A and B have almost identical environments except chiefly for sentences which contain both, we say they are synonyms... If A and B have some environments in common and some not... we say that they have different meanings, the amount of meaning difference corresponding roughly to the amount of difference in their environments.” (Harris 1970, sid. 786)

Harris menar att det är omöjligt att få mer än en grov uppskattning om ett ords vanligaste kontext, men att det *är* möjligt att mäta graden av likhet mellan två ords individuella kontexter. Detta innebär att det går att göra mer precisa uttalanden om ords likhet kontra andra ord. (Harris 1970)

Denna hypotes ligger till grund för den typ av vektorsemantisk analys som detta arbete bygger på. Det är en analysform där termernas kontexter utgör termernas betydelsebärande egenskaper. Detta är grunden för detta arbetes moderna system som ser text som behållare av ord och dessa ord, i det sammanhang de uppträder i, som indikatorer på textens innehåll eller begrepp.

## 2.4 Mening

För att kunna använda metoden som summeras ovan som en modell för semantisk likhet bör man först definiera begreppet likhet. Likhet är starkt ihopkopplat med mening, om två termer *menar* samma sak (till exempel ”ungkarl” och ”ogift man”) ser man dem som *lika*. Vidare bör man fråga sig varför den ovan beskrivna metoden är rimlig i sin modellering av mening.

Ett stort antal kända forskare har arbetat med denna fråga varav tre av dessa är psykologerna Burgess, Lund och Atchley. Dessa menar att de vektorer som skapas som termrepresentation är semantiska i sin natur samt att den vektorrymdsmodell som bygger upp hela den textuella representationen fungerar som en plausibel modell för semantiskt minne. (Burgess et al. 1995)

Burgess et. al. skapar en matris av de vektorer som definierats genom relativ ordförekomst på ord-ord nivå, det vill säga att matrisen får storleken  $v \times v$ , där  $v$  är den analyserade textsamlingens vokabulärstorlek. Hypotesen som ställs upp i denna undersökning är att denna matris på ord-ord nivå verkligen bär på semantisk information. I sina experiment med sin experimentmodell vid namn HAL (Hyperspace Analogue to Language) tar Burgess et. al. en mängd mänskliga

försökspersoner vilka får koppla samman ordpar<sup>3</sup>. Resultaten från de mänskliga klassifierarna tangerades sedan av den vektorsemantiska modell de skapat.

Det går givetvis att diskutera huruvida den distributionella fördelningen av ord verkligen återger, eller representerar, meningen i det ord eller den text som analyseras. Detta kommer att beröras vidare i sektion 2.4. Faktum kvarstår dock att de experiment som utfördes av Burgess et. al. gav resultat som var fullt jämförbara med mänskliga klassifiorare för de specifika experiment som utfördes. Det som framgår av detta är att termers distribution på något plan speglar den mänskliga uppfattningen av vad som utgör språklig mening och likhet i text. Detta tänkande, att man kan finna insikt om ords mening genom att studera statistiken kring deras förekomster, är inte en tanke utformad av dessa forskare utan är en hypotes som har studerats en längre tid inom modern lingvistik.

Vad är då egentligen mening? Är meningen någon form av "mentalt objekt" som endast står att finna i sinnet hos den egentliga språkutövaren, eller existerar meningen i språket i texten? Det är ingen lätt fråga att besvara vad mening är, och det är ingen fråga som kommer att försöka besvaras i och med denna uppsats. Detta arbete fokuserar på Harris hypotes. Denna hypotes ger oss att det är i den textuella distributionen av ord som man kan hitta och analysera *symptomen* av mening, med vilket Harris vill säga det språkliga användandet.

De vektorrymmodeller som experimentet nyttjar behöver inte tillgång till någon förståelse om, eller instruktion kring, vad mening innebär. Det enda antagandet som behöver göras är att om ord förekommer i samma eller liknande kontexter har de drag som tyder på semantisk likhet. Detta uttrycker inte vad mening är, utan fungerar enligt distributionshypotesen som en indikator på den textuella meningen.

## 2.5 Begreppsrepresentation

Ordrepresentation väljer att representera text på ett naivt sätt. Det finns ingen mekanism för att ta hänsyn till textens ordföljd, ordval eller de textuella resultaten av grammatiska regler. Ordrepresentation använder ett konstraintuitivt s.k. påsbegrepp där ordens förekomst i texten accepteras som markörer för mening. Det kallas påsbegrepp då dessa ords information kan ses som en påse av ord där förekomststatistiken är helt oberoende av de andra orden i det analyserade dokumentet. Det torde vara uppenbart att språket i sig är mer komplext än så.

Det är med detta i åtanke som metoder för begreppsrepresentation har utvecklats. I och med den av Zellig Harris skapade distributionshypotesen (se sektion 2.3) finns det ett verktyg för att mäta ords relativa likhet mot varandra med stöd i ordens distribution.

För att skapa en begreppsrepresentation behöver man först skapa ett högdimensionellt semantiskt rum. Detta rum representeras av en matris av samförekomststatistik. Storleken på denna matris sätts vanligen till den kvadrerade vokabulären eller alternativt vokabulären gånger mängden av dokument i den textmängd som skall analyseras. Skillnaden mellan dessa två strategier är i vilken kontext man väljer att tillämpa distributionshypotesen. Är man mer intresserad av ords distribution kontra andra ord eller är det mer relevant att analysera

---

<sup>3</sup>Ett exempel på ett av dessa ordpar var att BREAD skulle kopplas samman med ett av ett antal andra ord. Bland dessa ord återfanns bl.a. BUTTER och FLOOR.

denna distribution kontra vilka dokument dessa ord figurerar i? Olika modeller väljer olika strategier i denna fråga.

### 2.5.1 Från ord till ord

Den enda aspekten av textuell information som används när man skapar vektorrum med ordrepresentation är ren ordförekomst. När man beaktar språket komplexitet så inser man snart att språklig information färdas på många fler sätt än endast ordförekomst.

Detta blir mer uppenbart om man som exempel ser på polysema ord. Ta till exempel ordet "bad" som har olika mening i en kontext där texten refererar till kyrkliga aktiviteter mot en kontext där texten refererar till tvättning i stora mängder vatten. Orsaken till att ordet "bad" har olika mening i dessa två kontexter är inte någon intrinsikal egenskap hos ordet "bad" utan snarare hur detta ord används. Det vill säga att det inte är så att själva ordets semantiska egenskaper förändras utan att kontexten ändras, och det är med denna kontextuella förändringen som ordets mening förändras.

Dock har den vektorsemantiska metoden i stort vissa svagheter som inga varianter kan överbrygga. Hinrich Schütze beskriver ett av dessa problem i sin artikel *Dimensions of Meaning*:

It is possible to defeat this scheme by describing a content exclusively using words that normally express unrelated thoughts. But such situations are expected to be rare. (Schütze 1992)

### 2.5.2 Metoder för begreppsrepresentation

Det finns ett antal kända modeller för att skapa begreppsrepresentationer. Med *Begreppsrepresentation* menas de modeller som skapar sin representation baserat på den kontext ord förekommer i den analyserade textsamlingens dokument. Jag kommer innan jag introducerar den metod som används i denna uppsats huvudexperiment gå igenom tre andra modeller för begreppsrepresentation.

**Latent Semantisk Analys (LSA)** skapades av psykologen Landauer och datavetaren Dumais. Det är en dokumentbaserad metod och som sådan använder den sig inte av någon information angående ordföljd eller semantiska relationer. LSA skapar ett semantiskt rum i storleksordningen 50–1500 dimensioner från den ursprungsmatris som initialt har storleken  $v \times d$  där  $v$  är textsamlingens vokabulärstorlek och  $d$  är antalet dokument i textsamlingen. Detta sker genom en dimensionsreducerande process som kallas *Singular Value Decomposition* (SVD). LSAs skapare menar att det är fördelaktigt att utföra denna reduktion från en stor och datamaskinellt otymplig matris ner till den mindre matrisen  $v \times r$  där  $v$  är textsamlingens vokabulärstorlek och  $r$  är det reducerade rummet i storleksordningen 50–1500 dimensioner. Författarna anser att denna reduktion ger modellen en bättre approximation till den mänskliga kognitiva förmågan än vad den oreducerade datan skulle kunna göra. (Landauer, Foltz & Laham 1998).

SVD, som är en form av faktoranalys<sup>4</sup>, fungerar genom att dela upp den

---

<sup>4</sup>Faktoranalys är en beräkningsmetod som söker att försöka förklara korrelationen mellan ett antal observerade variabler med hjälp av ett mindre antal bakomliggande tänkta variabler, så kallade faktorer.

ursprungliga matrisen i tre nya delmatriser. Av dessa nya delar är det första en tabell med härledda faktorvärden från ursprungsmatrisens rader, den andra beskriver dess kolumner på samma sätt och den tredje beskriver den diagonala matrisen på ett sådant sätt att om det tre delarna skulle multipliceras ihop skulle ursprungsmatrisen återskapas. Genom att radera de minsta värdena i denna diagonala matris utförs dimensionsreduktionen.

Denna teknik har i experiment gett utmärkta resultat som speglar mänskliga. I Landauer & Dumais (1997) gör för författarna tester med LSA på det så kallade TOEFL-testet<sup>5</sup> där LSA får resultat fullt jämförbara med det mänskliga medelvärdet på sagda test (LSA åstakom 64,4% då det mänskliga medelvärdet låg på 64,5%). I jämförelse är detta ett resultat som skulle ge tillträde till ett stort antal högskolor i USA och Kanada för en student som inte har engelska som modersmål.

**Hyperspace Analogue to Language (HAL)** utvecklades av psykologerna Burgess, Lund och Atchley. De väljer att använda en träningsmetod som använder fönsterträning av storleken 10 ord i ett försök att lägga sig mellan LSAs dokumentträning och den mer strikta varianterna av fönsterträning med storlekar omkring ett ord. Dokument- och fönsterträning presenteras ingående nedan i sektion 2.6. HAL analyserar således termer mot termer vilket innebär att den matris som skapas får storleken  $v \times v$  där  $v$  är textsamlingens vokabulärstorlek.

För att göra arbetet med matrisen mer hanterbart blir Burgess et. al. likt Landauer et. al. tvungna att använda ett dimensionsreducerande steg. Denna reduktion fungerar genom att man helt enkelt förkastar de kolumner i matrisen som har lägst värden, det vill säga innehåller minst information. Med denna metod som författarna kallar sin *kolumnvariansmetod*, reducerar författarna i sitt experiment sitt rum nästan 700-falt ner till den mer hanterbara storleken av 200 dimensioner. Burgess et. al. genomför experiment, varav ett redan sammanfattats i sektion 2.4, med goda resultat. (Burgess et al. 1995)

**Slumpindexering (Random Indexing)** har utvecklats på SICS av Magnus Sahlgren med stöd av Pentti Kanervas arbete med glesa minnesrepresentationer, ett arbete som är besläktat med Samuel Kaskis arbete med *Random Mapping* (Kaski 1998). Grundtanken är att tilldela termer, eller dokument, placeringar i ett vektorrum som skapas med hjälp av *indexvektorer*. Dessa vektorer fungerar som en markör för ordets position i rummet genom att uttrycka en riktning i en eller flera dimensioner. Genom att tilldela varje ord, eller dokument, en unik sådan vektor till varje dokument strävar man efter att sprida dessa vektorer homogent i det semantiska rum man skapat. När detta steg är fullbordat skapar man en träningsstrategi för att bygga upp den egentliga representationen. Detta träningsmoment kan utformas på flera sätt (två vanliga strategier presenteras i sektion 2.6) men dess syfte är alltid det samma: att bygga upp en jämförande statistik över alla termers distribution till alla andra termer eller dokument med hjälp av termernas indexvektorer som referenser. Denna distributionella information lagras, för varje ord, i en ny vektor som kallas termens

---

<sup>5</sup>Test Of English as a Foreign Language

*kontextvektor*. Denna vektor innehåller en ny position som speglar ordets distribution i texten. Detta träningsmoment innebär att de vektorerna man skapar kommer att förändras beroende på vilken kontext de förekommer i. Dessa vektorer kommer att konvergera mot sina respektive termers generella distribution i texten med mängden av material som modellen utsätts för. Metodens stabilitet och precision är bunden till att uppnå en viss massa av träningstext. Den största styrkan med slumpindexering är att den är helt *inkrementell*. Men detta menas att den dimensionsreducerande process som används i LSA eller HAL inte förekommer på samma sätt som i dessa två modeller. Både SVD och HALs kolumnvariens "kollapsar" det semantiska rum man tidigare har byggt upp för att skapa en mindre modell över språket. Det är denna modell som används i en senare ev. textkategoriseringsprocess. Att addera ny information in i en sådan modell är inte helt omöjligt men väldigt tidsineffektivt. Slumpindexering har sitt dimensionsreduktionssteg inbyggt i skapandet av vektorrummet. Detta sker på det sättet att indexvektorerna representeras med ett antal positiva och negativa värden som slumpartat sprids i rummet. Detta leder till att man kan skapa ett stort antal unika indexvektorer i ett rum med mycket lägre dimensionaltitet än vokabulärstorleken. Eftersom metoden är helt inkrementell kan man extrahera preliminära resultat redan efter man analyserat en bråkdel av sin textsamling. En annan fördel med detta är att det datamaskinellt krävande dimensionsreduktionssteget tas bort. (Sahlgren n.d.)

I en upprepning av Landauer och Dumais TOEFL-experiment har slumpindexering lyckats prestera resultat på 72%, vilket är 7,6% bättre än LSAs resultat. Dessa resultat uppnåddes med hjälp av att endast analysera ordens stammar men även med tester med helt obehandlade ord lyckas prestera bra resultat. Slumpindexering presterade då nästan 2% bättre än LSA. (Karlgrén & Sahlgren 2001)

Beakta följande tankeexempel som en illustration på hur vektorrum byggs upp och förändras med en information som den byggs upp av. Detta exempel skulle kunna utföras med någon av de tre metoder som beskrivits ovan. Antag att vi skapar ett vektorrum över en textsamling kring flygning och flygteknik. Detta rum innehåller en stor mängd dokument samlade relativa ordfrekvenser med ett större antal flygtekniska termer. Efter att vi utfört experiment och diverse arbetsuppgifter med detta rum inser vi att vi behöver utöka detta rum med fler texter. Om vi nu utökar detta vektorrum med en större textsamling om till exempel glasstillverkning så kommer de allra flesta av de flygtekniska termernas kontextvektorer att ligga fast och inneha nästintill samma värden. Detta beror givetvis på att vi har slagits samman två så disparata ämnen. Det finns inga referenser till våra flygtekniska termer i dessa nya texter och således förändras inte kontexten i vilken dessa ord förekommer i. Det kommer dock att finnas vissa få överlappande termer, nämligen snittet i vokabulären mellan de två textsamlingarna. I detta fall skulle ett sådant ord kunna vara *isbildning*<sup>6</sup>. Dessa ord kommer då att förändras med den nya informationen och få någon form av medelvärde av ordets mening i dessa två skilda kontexter.

---

<sup>6</sup>Givetvis kommer snittet vara mycket större än bara ett fåtal ord, men i detta tankeexempel förutsätts att endast de mer tekniska facktermerna för varje ämnesområde, de potentiella klassorden, är de relevanta.

## 2.6 Kontextuella modeller

Det finns flera träningsmetoder man kan använda sig av när man skapar ett vektorrum. Ovan kunde man se hur enkla ordförekomster gick att använda, men det går även att se på ords kontexter som data för vektorrummet. I de följande experimenten har, som påtalats ovan, valet gjorts att använda fönsterträning som den huvudsakliga träningsmetod för att ge termerna distributionell information. Även ett specialfall av fönsterträning, som kallas dokumentträning, kommer att användas i experimenten. Dessa metoder beskrivs nedan.

### 2.6.1 Fönsterträning

Fönsterträning är en metod som väl stämmer överens med hur distributionshypotesen beskriver meningsrelationer i språket. Med hjälp av fönsterträning kan man se till att det är ords kontext som ligger till grund för hur dessa ords kontextvektorer kommer att se ut. Fönsterträning som metod har två variabler: fönstrets storlek ( $s$ ), som beskriver hur många av de direkt närliggande orden som skall beaktas i konstruktionen av fokalordets kontextvektor samt vikningsschemat som beskriver till hur stor del dessa ord skall beaktas när man bygger sagda vektor.

I fönsterträning stegar man igenom det dokument eller den textsamling, som skall ligga till grund för representationen, ord för ord. Det ord man för tillfället undersöker kallas *fokalordet*. När man undersöker detta fokalord ser systemet till så många av de direkt närliggande orden som fönsterstorleken anvisar och tar hänsyn till den för att bygga en kontextrepresentation för fokalordet. Nedan ges en lite närmare genomgång på en algoritm för detta. I figur 2 nedan går det att utläsa hur de olika vikterna fördelas i fönsterstorlekar  $s$  som spänner från 2 till 8 ord med vikningsschemat  $2^{1-d}$  där  $d$  är avståndet till fokalordet  $f$ . Fördelarna med detta vikningsschema förklaras nedan i sektion 3.3. Värdena motsvarar till hur stor del grannordet skall verka på fokalordets representation.

	”That	rabbit’s	got	a	<i>vicious</i>	streak	a	mile	wide”
$s = 2$ :				1	$f$	1			
$s = 4$ :			0,5	1	$f$	1	0,5		
$s = 6$ :		0,25	0,5	1	$f$	1	0,5	0,25	
$s = 8$ :	0,125	0,25	0,5	1	$f$	1	0,5	0,25	0,125

Figur 2: Exempel på fönsterträning med varierande fönsterstorlek

### 2.6.2 Dokumentträning

Dokumentträning är ett specialfall av fönsterträning där fönstrets storlek alltid täcker in den resterande del av det aktuella dokumentet man analyserar, både före och efter fokalordet. Detta innebär att fönstret inte alltid blir symmetriskt med avseende på fönstrets storlek innan och efter fokalordet. Denna träningsmetod skiljer sig ifrån fönsterträning i att den arbetar vidare ifrån förutsättningen att hela det aktuella dokumentet man analyserar är relevant för de ord som förekommer i dokumentet. Med andra ord förutsätter träningsmetoden

att dokumentet behandlar ett singulärt koncept. I dokumentträning förekommer ingen termviktning utan varje term bedöms lika viktiga. Detta viktningsschema tar vidare ingen hänsyn till ordföljd eller till andra semantiska relationer.

## 2.7 Unär indexering

*Unär indexering* är en variant av slumpindexering där dimensionsreduceringssteget har plockats bort. Skapandet av vektorrummet blir då lite annorlunda. Där man i slumpindexering ger varje term en riktning bestående av ett antal positiva och negativa värden som slumpartat sprids ut i vektorrummet är upplägget i *Unär indexering* linjärt. Varje term representeras i sin indexvektor med endast ett värde i en dimension. Eftersom vektorrummens dimensionalitet sätts till textsamlingens vokabulärstorlek finns det exakt en dimension per unik term. Dessa vektorer associeras med vektorrummens ortogonala<sup>7</sup> basvektorer.

Fördelen med detta upplägg är att eftersom ingen dimensionsreduktion görs blir inte den störning som en sådan operation medför inkluderad i resultatet. Denna störning utgörs av *brus* som introduceras i datan då man gör den samlade informationen mer lågupplöst då man minskar antalet dimensioner som representerar texten. Modellens nackdel är dock att utan dimensionsreduktion blir de skapade vektormatriserna snabbt oerhört stora. En textsamling med 50 000 unika ord i 25 000 olika dokument skulle representeras i en matris av storleken  $50\,000 \times 25\,000$  i LSA innan dimensionsreduktion och skulle kunna representeras som  $50\,000 \times 2000$  i Slumpindexering. *Unär indexering* kräver dock att storleken blir  $50\,000 \times 50\,000$ . I detta fall blir matrisen 25 gånger större när den behandlas med *unär indexering* i jämförelse med Slumpindexering. Detta gör modellen mindre lämpad om man skall analysera större datamängder men inte har tillgång till likvärdigt större datamaskinell beräkningskapacitet.

*Unär indexering* har alla de positiva egenskaperna av Slumpindexering vad gäller inkrementaliteten, vilket gör den väldigt adaptiv till ny information och gör den enkel att expandera med mer text om sådan finns tillgänglig. Dock innebär avsaknaden av ett dimensionsreducerande steg att *unär indexering* är att föredra om den textmängd man vill analysera är liten och/eller specialiserad. Avsaknaden av ett dimensionsreducerande steg gör att man slipper introducera brus in i modellen vilket gör den skördade samförekomststatistiken mer klar vilket givetvis sker på beskostnad av behandlingstid.

## 2.8 Analysverktyg

Experiment är utförda med analysverktyget GSDM (Guile Sparse Distributed Memory) som har tillhandahållits av SICS. Verktyget är utvecklat internt på SICS. Analysverktyget är skrivet i C för Unix och det har ett gränssnitt som används via en terminalapplikation. GSDM är en Guileolk ursprungligen skriven av Anders Holst på SICS för att implementera Pentti Kanervas idéer om hur det mänskliga minnet fungerar. Det har funnit en ny användning i att modellera vektorrum i denna uppsats experiment och många tidigare experiment på SICS. GSDM tar instruktioner skrivna i programmeringspaket Guile som är en dialekt av Scheme. Scheme i sin tur är en variant av Lisp.

---

<sup>7</sup>Ett vektorrum med  $n$  dimensioner har exakt  $n$  ortogonala basvektorer. Dessa vektorer har endast riktning i en axel i rummet.

## 3 Experiment

### 3.1 Data

Textsamling bestod av 2791 dokument vilka i sin tur bestod utav dokumentens titel och sammandrag (abstract). Dock existerade det vissa fall där dokumenten endast bestod utav en titel. Till dessa dokument hörde en mängd fördefinierade kategorier, eller klasser. Dessa klasser var 1423 till antalet. Varje dokument var tilldelade ett antal av dessa klasser, i medeltal cirka 20 stycken. I figur 3 nedan har ett dokument ur datamängden återgivits. Förkortningarna TI, AB samt CT står för Titel, Sammandrag (Abstract) samt Klasser. Med Klasser menas den lista av klasser dokumentet har tilldelats av mänskliga klassificerare innan dessa experiment.

- TI Polyelectrolyte templated polyaniline - film morphology and conductivity
- AB A simple synthesis protocol for stable aqueous colloidal solutions of poly(4-styrenesulfonate) templated polyaniline is developed. The electrical conductivity and submicro/nano features observed in their spin-coated films are shown to be correlated to the polyelectrolyte template molecular weight. This demonstrates the utility of the latter as a new design element for conducting polymer films. (orig.)
- CT Colloids; Conductivity, Electrical; Electron microscopy; ESCA; Films; IR spectroscopy; Molecular weight; Morphology; Polyaniline; Polyelectrolytes; Polystyrene, Sulfonated; Spin coating; UV Spectroscopy

Figur 3: Ett dokument med tillhörande klasser före normalisering

### 3.2 Förprocessering

Textsamlingen splittrades upp i två delar, en som bestod utav titlarna konkatenerade med sammanfattningarna samt en som bestod utav klasserna för varje dokument. Denna uppdelning skapar således två nya texter, en innehållandes 2791 titlar med tillhörande sammanfattningar samt en med 2791 klasslistor. Dessa två texter kallas hädanefter *texten* respektive *nycklarna*.

#### 3.2.1 Normalisering

Text och nycklarna normaliserades först genom att skriva om versaler till gemener. Detta för att GSDM (se sektion 2.8) annars skulle separera ord med till exempel en versal begynnelsebokstav som en separat ordform. Vidare normaliserades nycklarna genom att ta bort alla förekomster av klasser bland nycklarna förutom förekomst i *texten*. Detta gjordes för att stävja *sparse-data* problemet, som i det här fallet innebär att det inte går att uttala sig om ett ord som inte har någon kontextinformation associerat till sig. Således går det inte att med hjälp av distributionshypotesen att utröna något om hur detta ord förhåller sig till resten av de ord som utgör den aktuella texten. Med andra ord "Vad man inte kan tala om, därom måste man tåga".

Vidare fanns det ett litet antal klasser bland nycklarna som bestod av ett huvudord med en specificerare på formen  $(m_1, m_2)$ . I dessa fall togs specificeraren helt bort och ersattes med huvudordet  $(m_1)$  då dessa kombinationer av ord

förekom ytterst sällan, alternativt inte alls, i texten. I enlighet med samma motivation som ovan reducerades dessa sammansättningar till deras huvudord. I alla fall var dessa huvudord termer som var väl representerade i texten. Detta tillät mig att behandla dessa ord i enlighet med distributionshypotesen i stället för att tvingas ta bort dem liksom icke förekommande termer. Ett exempel från figur 3 är *Conductivity, Electrical* som i figur 4 har normaliserats till *conductivity*.

Ytterligare ett problem var att vissa av klasserna bland nycklarna var flerords-termer (hädanefter kallade  $n$ -gram). Detta var ett problem eftersom analysverktyget, GSDM, fokuserar på singulära ord. För att komma förbi detta problem, utan att ersätta dessa  $n$ -gram med en sammanfattande term; alternativt exkludera dessa klasser helt och hållet, användes en annan infallsvinkel. En ny representation skapades av alla  $n$ -gram bland klasserna samt alla dess förekomster i dokumenten. Dessa ord skrevs om till sammansatta ord på formen  $n_1n_2n_n$ . Detta löser problemet utan att på något sätt introducerade någon form av betydelseförändring i texten, eller bland klasserna. Ett faktiskt exempel på denna modifikation från texten är ordet *Electron microscopy* som för varje förekomst, i texten såväl som i nycklarna, skrevs om till *electronmicroscopy*. Detta tillåter mig att behandla de klasser som var längre än ett ord helt i enlighet med distributionshypotesen.

Ett sista problem med texten var att analysverktyget GSDM bortser från alla ord innehållandes tecken '\_', ' ' samt '/'. I normaliseringen togs samtliga av dessa tecken bort. Ord som innehöll ett sådant tecken konkatenerades ihop till ett ord. Som exempel normaliserades *submicro/nano* till *submicronano*.

I figur 4 nedan återges dokumentet från figur 3 efter att de ovan beskrivna normaliseringarna genomförts.

polyelectrolyte templated polyaniline film morphology and conductivity  
a simple synthesis protocol for stable aqueous colloidal solutions of  
poly(4styrenesulfonate) templated polyaniline is developed. the electrical  
conductivity and submicronano features observed in their spincoated  
films are shown to be correlated to the polyelectrolyte template molecular-  
weight. this demonstrates the utility of the latter as a new design element  
for conducting polymer films. (orig.)

colloids conductivity electronmicroscopy esca films irspectroscopy mo-  
lecularweight morphology polyaniline polyelectrolytes polystyrene spin-  
coating uvspectroscopy

Figur 4: Ett dokument med tillhörande klasser efter normalisering

### 3.3 Experimentutförande

Experimenten gick till på följande sätt. Ett högdimensionellt semantiskt rum skapades, där dimensionaliteten sattes till vokabulärens storlek. I detta fall fanns det 13672 unika ordformer efter normalisering, således skapades ett rum med sagt antal dimensioner. Efter det tilldelades det en unik vektor, kallad *indexvektorn*, i detta rum till varje ordform. Detta skapades enkelt genom att låta varje ord associeras med en egen dimension samt att låta varje ord få värdet +1 i dess indexvektor för denna dimension. Detta värde tilldelades orden för att för-

enkla de kommande beräkningarna. Denna indexvektor fungerar, som namnet antyder, likt en referens till ordets plats i vektorrummet.

När nu varje ordform är placerad i rummet behöver man skapa ett mått på ordets betydelse i jämförelse med de andra orden i vektorrummet. För att åstadkomma detta skapar man ytterligare en vektor för varje ord. Denna vektor kallas ordets *kontextvektor*. Det är denna vektor som kommer att innehålla information om ordets ordfrekvens relativt alla andra ord i rummet. Denna information tillförs ordets kontextvektor genom fönsterträning, se sektion 2.6 på sida 13. Detta innebär att när ett ord förekommer adderar man in de närliggande ordens indexvektorer. Detta sker för varje gång ordet förekommer och de respektive indexvektorerna viktas efter formeln  $2^{1-d}$  där  $d$  är avståndet till fokalordet. Detta viktningschema har visat sig vara ett fördelaktigt viktningsmönster för små fönster i storleken 4 – 6 av Sahlgren (2001).

När detta är gjort skapas en dokumentvektor  $\vec{d}$  genom att skapa en ny vektor och sedan addera in dokumentets samlade ords kontextvektorer in i  $\vec{d}$ . Dokumentvektorn  $\vec{d}$  representerar nu dokumentets *begrepp*. Detta har visat sig vara en fördelaktig representation av dokumentbegrepp i ett antal vektorsemantiska experiment i en sådan grad att det har blivit en standardmetod för att representera dokument av den storlek som förekommer i detta arbete.

### 3.3.1 En belysning av experimentmetoden med ett demonstrationsdokument

För att åskådliggöra denna process kommer ett reducerat exempel av hur denna metod realiserar presenteras steg för steg. Betänk dock att detta exempel är endast här för att belysa metoden. Den datamängd som används nedan är för liten för att kunna generera några som helst relevanta resultat, se därför detta avsnitt endast som en belysning av metodens tillvägagångssätt. Som exempel dokument för detta syfte, beakta följande utdrag ur dikten ”Det perfekta brottet” av Thomas Tidholm (1991):

*Ingen klarar av ett riktigt polisförhör. Och det vackraste en människa kan göra är att ge upp, fattig och syndig. Det vackraste en människa kan få vara med om är att slutligen bli överbevisad.*

Figur 5: Ett demonstrationsdokument

Texten normaliseras, i detta fall endast med avseende på versalerna. Texten läses in i GSDM som skapar ett flerdimensionellt semantiskt rum med lika många dimensioner som det finns unika ordformer. I detta fall har dokumentet 26 unika ordformer vilket innebär att ett rum med 26 dimensioner initialiseras. Detta innebär att alla ord i dokumentet tilldelas två vektorer. Den första vektorn kallas för ordets *indexvektor*. Denna vektor är en lägesbestämmande vektor och den tilldelas under initialiseringen en unik dimension i rummet. Detta innebär i sin förlängning, tillsammans med det faktum att det finns exakt 26 ordformer och således exakt 26 dimensioner, att alla ords *indexvektorer* sammantaget kommer att representera samtliga dimensioner i rummet. Varje ord tilldelas i och med att denna vektor skapas värdet +1 i sin tilldelade *indexvektor* för att dessa vektorer skall kunna beräknas på ett så enkelt sätt som möjligt. Resterande vektorer i vektorrummet har värdet 0. För att underlätta i förståelsen i detta exempel

Ord	Indexvektor	Kontextvektor
ingen	0+1	1+1
klaras	1+1	0+1 2+1
av	2+1	1+1 3+1
ett	3+1	2+1 4+1
riktigt	4+1	3+1 5+1
polisförhör	5+1	4+1 6+1
och	6+1	5+1 7+1 17+1 18+1
det	7+1	6+1 8+2 18+1
vackraste	8+1	7+2 9+2
en	9+1	8+2 10+2
människa	10+1	9+2 11+2
kan	11+1	10+1 12+1
göra	12+1	11+1 13+1
är	13+1	12+1 14+2 22+1
att	14+1	13+2 15+1 23+1
ge	15+1	14+1 16+1
upp	16+1	15+1 17+1
fattig	17+1	6+1 16+1
syndig	18+1	6+1 7+1
få	19+1	11+1 20+1
vara	20+1	19+1 21+1
med	21+1	20+1 22+1
om	22+1	21+1 13+1
slutligen	23+1	14+1 24+1
bli	24+1	23+1 25+1
överbevisad	25+1	24+1

Tabell 1: Ett litet semantiskt rum

kommer denna dimension att stämma överrens med förekomstordningen för orden. Dimensionerna numreras från 0 till 25 där dimension 0 associeras med ordet *Ingen* och dimension 25 associeras med ordet *överbevisad*. Den andra vektorn är *kontextvektorn* som kommer att skapas genom fönsterträning av dokumentet. För enkelhetens skull görs valet att träna detta dokument med en fönsterstorlek på 2, det vill säga man endast tittar på det ord som står närmast fokusordet på varje sida. Dessa ords förkomstvektor viktas efter formeln  $2^{1-d}$  (se sektion 3.3) och adderas in i det aktuella ordets kontextvektor. Detta innebär att vektorerna adderas in fullt ut då avståndet till fokalordet endast är 1 då fönsterstorleken är 2 (fönsterstorleken 2 innebär praktiskt ett fönster som innefattar ett ord på varje sida om fokalordet, vilket ger oss att  $d = 1$ ) Efter initialisering och träning får vektorerna de värden som visas i tabell 1 ovan. Vektorerna är representerade i tabellen på formen  $n^d + n^v$  där  $n^d$  är numret på dimensionen och  $n^v$  är vektorstorleken. Genom att addera ihop alla kontextvektorer för alla förekomster av ord skapas sedan en dokumentvektor för dokumentet. Denna vektor, som finns återgiven som figur 6 nedan, representerar nu alla dokumentets ords sammantagna kontextvektorer.

0+1 1+2 2+2 3+2 4+2 5+3 6+5 7+7 8+8 9+8 10+6 11+6 12+4 13+6  
14+6 15+3 16+2 17+3 18+4 19+1 20+2 21+2 22+3 23+3 24+2 25+1

Figur 6: Demonstrationsdokumentets dokumentvektor

### 3.3.2 Delexperiment

På texten utfördes det ett antal experiment för att utröna vilka parametrar var mest fördelaktiga i att skapa den bästa representationen för den aktuella texten. Som jämförelse mot konventionella metoder gjordes en ordrepresentation baserad på  $tf \times idf$  (se sektion 2.2). Utöver denna gjordes fem stycken ytterliga körningar på texten med unära vektorer. Det som skiljde dem åt var storleken på fönstret som användes vid fönsterträningen av ordens kontextvektorer. Den sista av dessa var en körning där träningsfönstret sattes till det aktuella dokumentet. Denna variant av fönsterträning, som kallas dokumentträning, täcker konsekvent in hela det aktuella dokumentet oavsett vilket fokordet är. Således var experimenten sex till antalet.

- Ordrepresentation med  $tf \times idf$ .
- Begreppsrepresentation med fönsterträning av storlek 2.
- Begreppsrepresentation med fönsterträning av storlek 4.
- Begreppsrepresentation med fönsterträning av storlek 6.
- Begreppsrepresentation med fönsterträning av storlek 8.
- Begreppsrepresentation med dokumentträning.

### 3.4 Utvärdering

I denna uppsats byggs dock inte en klassificerare för att utvärdera de resultat som skördats ur representationen utan arbetet fokuserar på hur väl representationen överrenstämmer med de fördefinierade dokumentnycklarna. Det är denna överensstämmelse som söks att utvärderas. Experiment som behandlar utvärdering av representationer med klassificerare har utförts av Sahlgren & Cöster (2004) med väl dokumenterade resultat. Denna uppsats skiljer sig ifrån dessa experiment genom att det är representationens integritet som är experimentets huvudfråga, inte hur väl olika klassificerare kan skörda information ur sagda representation. Denna uppgift är svårare än den som ställs inför Sahlgren & Cöster (2004) men på grund utav att detta arbete är ett uppdrag av en sådan typ att den data som erhållits är starkt begränsad har valet gjorts att inte applicera den lösning som används i Sahlgren & Cöster (2004). Datans karakteristika innebär att detta problem bör betraktas som ett nyckelordsextraktionsproblem.

När man skapar representationer vill man bygga dem runt den maximala mängd material man har tillgängligt. Detta är för att undvika *sparse-data* problemet, det vill säga problemet att i sin testmängd stöta på information som man inte påträffat tidigare och således inte har någon information om. Viktig information om aktuella klassers relativa ordfrekvens skulle då gå förlorad och representationen skulle då vara mindre representativ än den skulle kunna vara.

Den textsamling som har behandlats kommer från företaget INNOVEAS och inga möjligheter fanns att utöka dess storlek.

Som nämndes ovan bör detta betraktas som ett nyckelordsproblem, inte ett klassificeringsproblem, därför blir det inte meningsfullt att göra den klassiska datainlärningsseparationen mellan träningsdata och testdata. Detta beror på att metoden i grund är inkrementell. Det finns det inga skäl till att göra en dylik uppdelning, modellen är och kommer inte att vara statistiskt sluten modell, en 'svart låda', utan en inkrementell metod som med varje nytt dokument den analyserar förändras. Den modell av språket man skapar genom denna representation blir egentligen aldrig färdigtränad: modellen förändras och blir mer nyanserad för varje ytterligare dokument den ser, vilket innebär att modellen behöver så mycket material som möjligt för att kunna prestera så bra resultat som möjligt. Således är detta problem inte ett maskininlärningsproblem utan representationsgenerering, och med den begränsade datan i åtanke skulle det vara kontraproduktivt att ta en del av denna åt sidan.

Informationen om varje ord är lagrad i en lista där varje element representerar den relativa närheten till ett visst ord. Eftersom varje ord har en egen position i listan är grundstrukturer för varje ord densamma, en lista med lika många element som unika ordformer i texten där vissa positioner är associerad med ett värde. Dessa listor behandlas som *kontextvektorer* där  $k_i = (o_1, o_2 \dots o_n)$  där man låter förekomststatistiken för varje ordform  $o$  representera en egen dimension. När dessa vektorer analyseras med avseende på deras vektorlikhet mot varandra kan man uttala sig om hur lika dessa ord är sin textuella distribution mot varandra. Detta ger oss tillsammans med distributionshypotesen ett, genom vektorlikhet, lätt kvantifierbart mått på hur lika dessa ord är varandra.

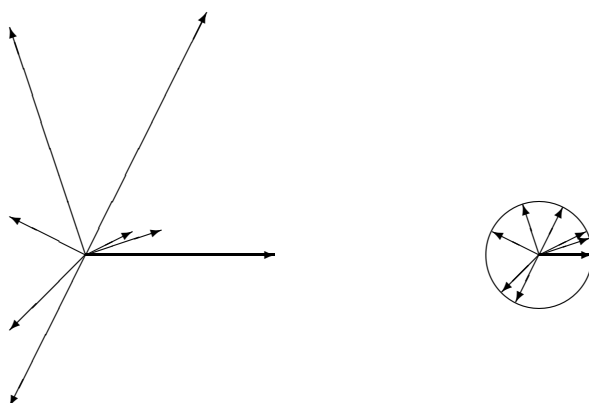
Det skapas således en lista  $l$  på godkända klasser för varje dokument. Denna lista innehåller den kompletta nyckellistan där varje ord är associerat med sin tränade kontextvektor. Denna lista är de begrepp som tilldelats varje dokument av mänskliga klassificerare. Dessa är i avseende på experimenten de *korrekta* begreppen för varje dokument, utvärderingen kommer att vara centrerad på att redovisa till vilken grad experimenten genererade samma begrepp som dessa.

### 3.4.1 Vektorlikhet

Att beräkna vektorlikhet är en matematiskt sett rättfram procedur. Det finns ett antal likhetsmått att nyttja sig av när man skall mäta likhet. En vanlig metod är beräkna skalärprodukten (se nedan) mellan vektorerna. I dessa experiment innefattar denna operation först en vektornormalisering vilken omvandlar samtliga vektorers olika magnituder till samma storlek (se fig. 7) för att motverka frekvens effekter. Med frekvens effekter menas en snedvridning av resultaten genom till exempel: om mer utstuderad information finns om de ord som förekommer ofta i texten, överrepresenteras dessa i resultaten. Detta är viktigt för att undvika situationer som till exempel den där två liknande ord inte märks ut som lika, enbart baserat på att det existerar mer information om det ena ordet.

Två av de vanligaste likhetsmåten för att beräkna vektorlikhet är euklidiskt avstånd samt kosinusmättet. Mättet för det euklidiska avståndet mellan två punkter i ett rum är:

$$d_E(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

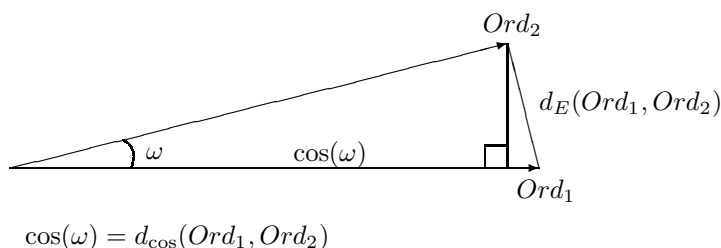


Figur 7: Ett enkelt vektorrum före och efter normalisering

Denna formel är dock inte normaliserande. Eftersom man bara arbetar med normaliserade data (enligt resonemang ovan) så kan man istället fokusera på vinkeln mellan riktningarna till vinkeln, eller med andra ord, vinkeln mellan vektorerna. Ett lättberäknat mått som beror på vinkeln mellan två riktningar är just kosinus, som går från ett, vid samma riktningar, till noll, vid ortogonala riktningar. Beräkning av kosinus görs med följande formel:

$$d_{\cos}(x, y) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

Nämnummern i formeln är en längdnormalisering och täljaren är skalärprodukten. Notera i figur 8 nedan en belysning av vad kosinusvärdet respektive det euklidiska avståndet beräknar, med hjälp av två exempelvektorer. Måttet på det euklidiska avståndet är likt en invers på kosinusmåttet. Kosinusmåttet är som störst vid liknande vinklar och är således ett närhetsmått och det euklidiska avståndet blir som störst vid olika vinklar då det är ett avståndsmått. I dessa experiment har valet dock gjorts att beräkna ord och dokumentvektors likhet mot varandra med hjälp av kosinusmåttet. Detta endast för beräkningsmetodens enkelhet och effektivitet.



Figur 8: Kosinusvärde och Euklidiskt avstånd mellan två vektorer

### 3.4.2 Utvärderingsmetod

Dokumenterna utvärderades genom att göra en begreppsextraktion av dokumentvektorn  $\vec{d}$  för varje dokument. Orden från den färdigställda klasslistan  $l$  rangordnades med avseende på kontextvektorer genom vektorlikhet mot  $\vec{d}$  varvid

precision och täckning beräknades. Tre huvudsakliga metoder användes sedan i utvärderingen.

1. En rak utvärdering, hädanefter kallad *strikt* utvärdering: där man genom vektorlikhet tar ut de  $n$  antal mest liknande klasselementen där  $n =$  antalet förtilldelade klasser till dokumentet. En jämförelse görs sedan hur många av de uttagna orden överrensstämmer med dokumentets verkliga klasser. I detta fall blir precisionen lika med täckningen.
2. En singularutvärdering, hädanefter kallad *primär* utvärdering: där man tar ut det ord från listan  $l$  som är det mest överrenstämmande med dokumentvektorn  $\vec{d}$ . Detta ord testas sedan om huruvida det finns med bland de klasser som tilldelats dokumentet. Denna metod blir således precisions-effektiv.
3. En generös utvärdering, hädanefter kallad *bred* utvärdering: där man genom vektorlikhet tar ut de  $n$  antal mest liknande klasselementen där  $n =$  det dubbla antalet förtilldelade klasser till dokumentet, dock minst 20 klasser. En jämförelse görs sedan hur många av de uttagna orden överrensstämmer med dokumentets verkliga klasser. Denna metod är således täckningseffektiv.

## 4 Resultat

Resultaten från ordrepresentationen summeras nedan i tabell 2. De beräknade resultaten är andelen korrekta klasser genererade av representationen. Tabell 3 summerar resultaten av experimenten med begreppsrepresentation. Resultaten är beräknade med avseende på precision (antalet korrekt genererade klasser genom mängden genererade klasser) och täckning (antalet korrekt genererade klasser genom antalet fördefinierade klasser).

	Normal		Primär	
	Precision	Täckning	Precision	Täckning
$tf \times idf$	16,34%	47,35%	66,07%	3,65%

Tabell 2: Ordrepresentationens resultat

	Strikt		Primär		Bred	
	Precision	Täckning	Precision	Täckning	Precision	Täckning
1+1	10,72%		27,95%	1,43%	7,54%	15,15%
2+2	12,52%		29,38%	<b>1,52%</b>	9,84%	19,76%
3+3	12,38%		<b>29,60%</b>	1,22%	10,07%	20,24%
4+4	12,30%		29,42%	<b>1,52%</b>	10,12%	20,33%
dok.	<b>13,95%</b>		17,45%	0,86%	<b>10,59%</b>	<b>21,23%</b>

Tabell 3: Begreppsrepresentationens resultat

Som man kan se ger den dokumenttränade representationen något bättre resultat än de fönstertränade i den strikta och breda utvärderingen. I den primära utvärderingen gav dock fönsterträningen med fönsterstorleken 3+3 den högsta graden av korrelation.

En mycket intressant aspekt av resultaten är att för den stora majoriteten av dokumenten ger experimenten en lista där de första orden alltid är desamma. Det förekommer en begränsad omkastning av ordning bland dessa ord i vissa fall men den är oftast lika. Individuella dokumentens unika 'fingeravtryck' infinner sig tidigast efter 5 ord men oftast efter fler än så. Detta är ett klart tecken på att det semantiska rummet är 'ihopklumpat'. Med detta vill säga att dokumentvektorerna alternativt klassvektorerna ligger väldigt tätt intill varandra.

Detta innebär även att den primärutvärdering som gjorts av dokumentens högst korrelerade ontologiord är väldigt statistisk. På grund av de 'ihopklumpade' resultaten är det högst korrelerade ordet oftast det samma.

## 5 Diskussion

Experimenten har visat att den mängd specialiserade sammandrag som användes för att skapa representationer inte var tillräckligt omfattande. Det semantiska rummet som utgör representationen ter sig sammandraget och likriktat vilket genererar överväldigande likformiga resultat vid utvärdering. Det är detta arbetes slutsats att detta inte bara har att göra med den relativt lilla datamängd som har använts utan även med dokumentens form, vilken oftast tenderar till att vara uppräknande och så tekniska att det befinner sig i gränslandet av vad som kan anses vara naturligt språk. Faktumet att dokumentrepresentationen får något bättre resultat i de två mest omfattande utvärderingarna talar för detta. Således är det arbetets slutsats att de resultat som detta arbete har genererat beror på en kombination av följande orsaker:

- Dokumenten är väldigt kompakt skrivna och söker att få med hela dokumentens syfte och mål i några få rader. I många fall liknar även dokumenten en dispositionslik uppräknning av allt det som det egentliga dokumentet behandlar. Många av klasserna finns med men befinner sig i en 'styldad' kontext vilket aktivt leder till att resultaten blir lidande.
- Textmängden är liten vilket leder till att själva träningsmomentet för arbetet blir undermåligt. Till detta läggs det faktum att sammandrag oftast är väldigt koncisa och oexpansiva i avseende på textmassa.
- Texten är väldigt tekniskt skriven och fackordintensiv, vilket bidrar till att mängden lågfrekvensord blir större vilket minskar mängden träningsinformation per unik ordform.
- Dokumenten är ofta rika på klasser men vissa basala, dock centrala, klasser som t.ex. *Electron microscopy* förekommer relativt sällan i dokumenten men är en av de mest förekommande klasstermerna (*Electron microscopy* är den näst mest förekommande klassen och är klassord i 1325 dokumentens klasslistor, dock förekommer ordet totalt endast 343 gånger i de 2791 dokumenten).
- Det är slutligen odefinierat på vilket sätt dessa dokument har tilldelats sina klasser. Är det endast en person som tilldelat dessa klasser eller är det flera? Det är lätt att ifrågasätta de existerande klassernas integritet och deras grad av konsekvens.

Det är således författarens slutsats att det finns problem med att endast använda titlar och sammandrag i en analys likt den som utförts ovan. Denna slutsats stärks även av Dr. Anette Hulth's resultat, vilka beskrivs i hennes nyligen framlagda doktorsavhandling på Stockholms Universitet. (Hulth 2004)

Det är vidare möjligt att på grund av dokumentens uppräknande natur är dessa resultat en effekt av sekundära frekvenseffekter inom dokumenten som kan ha lett till en viss överrepresentation av vissa termer. Även om en frekvensnormalisering utförs vid kosinusberäkningen kan sekundära frekvenseffekter förekomma då dokumentvektorn för ett enskilt dokument skapas. Eftersom denna vektor skapas av dokumentets kontextvektorer innan dessa normaliseras blir denna vektors riktning partisk till ofta förekommande ord i dokumentet. Denna effekt är dock något som man eftersträvar eftersom man vill behandla kontexter

av ord, i detta fall dokument, inte bara enskilda ord (se sektion 2.6). Man vill att dokumentvektorn skall dras åt hela dokumentets *begrepp* men om texten som behandlas repeterar en eller en viss typ av term ofta kommer denna metod att även bli något av en belastning. Detta problem kan te sig litet, speciellt i dokument runt 50–60 ord som dessa experiment är utförda på, det är dock en faktor vilken kan ha förstärkt problemet.

## 6 Sammanfattning

Med hjälp av två vektorsemantiska metoder, klassisk ordrepresentation samt begreppsrepresentation har detta arbete försökt skapa en användbar semantisk representation av en datamängd för att undersöka möjligheterna till att skapa ett analysverktyg för textkategorisering. Dock har arbetet plågats av dåliga resultat vilket är produkten av att träningsmaterialet var undermåligt både i kvalité och kvantitet. Detta arbete har visat att även en god representation till trots, kommer mager indata att leda till dåliga resultat. Experimenten med ordrepresentation har genererat mycket bättre resultat än begreppsrepresentationen. Det är detta arbetes slutsats att det beror på de dåliga grundförutsättningarna vad gäller textmängd samt kontextuella kvalitén på denna text.

I ett komplett dokument, skulle texten vara mer normaliserad i avseende på klasserna vilket skulle leda till mer nyanserade semantiska vektorer för de relevanta klasserna. Som exempel på detta kan åter nämnas klassen *Electron microscopy* som är en av de vanligaste klasserna i den aktuella datamängden. Dock är ordet underrepresenterat i sammanfattningarna eftersom ordet i sig är så grundläggande, och igenom sin basalitet — underförstådd, i dessa nanoteknologiska arbetens sammandrag.

## Referenser

- Burgess, C., Lund, K. & Atchley, R. A. (1995), Semantic and associative priming in high-dimensional semantic space, *i* 'Cognitive Science Proceedings, LEA'.
- Harris, Z. (1968), *Mathematical Structure of Language*, Interscience Publishers, New York.
- Harris, Z. S. (1970), Distributional structure, *i* H. Hiz, Z. S. Harris & H. M. Hoeningswald, eds, 'Papers in Structural and Transformational Linguistics', D. Reidel Publishing Company, ss. 775–794.
- Hulth, A. (2004), Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction, Doktorsavhandling, Stockholms universitet, Stockholm.
- Karlgren, J. & Sahlgren, M. (2001), From words to understanding, *i* Y. Uesaka, P. Kanerva & H. Asoh, eds, 'Foundations of Real-World Intelligence', CSLI Publications, ss. 294–308.
- Kaski, S. (1998), Dimensionality reduction by random mapping: Fast similarity computation for clustering, *i* 'Proceedings of IJCNN'98, International Joint Conference on Neural Networks', Vol. 1, ss. 413–418.
- Landauer, T. K. & Dumais, S. T. (1997), 'A solution to platos's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge', *Psychological Review* **104**(2), 211–240.
- Landauer, T. K., Foltz, P. W. & Laham, D. (1998), 'An introduction to latent semantic analysis', *Discourse Processes* **25**, 259–284.
- Robertson, S. E. & Sparck Jones, K. (1996), Simple, proven approaches to text retrieval, Technical Report 356, Computer Laboratory.
- Sahlgren, M. (2001), Vector-based semantic analysis: Representing word meanings based on random labels, *i* 'Proceedings of the Semantic Knowledge Acquisition and Categorisation Workshop at ESSLLI '01'.
- Sahlgren, M. (n.d.), Random indexing of words in narrow context windows for vector-based semantic analysis. Alessandro Lenci, Simonetta Montemagani, Vito Pirrelli, eds, Acquisition and Representation of Word Meaning: Theoretical and Computational Perspectives.
- Sahlgren, M. & Cöster, R. (2004), Using bag-of-concepts to improve the performance of support vector machines in text categorization, *i* 'Proceedings of the twenty-first International Conference on Computational Linguistics (COLING)'.
- Schütze, H. (1992), Dimensions of meaning, *i* 'Proceedings of Supercomputing '92'.
- Sebastini, F. (2002), 'Machine learning in automated text categorization', *ACM Computing Surveys* **34**(1), 1–47.
- Tidholm, T. (1991), *Friluftsliv i Strandområden*, Wahlström & Widstrand.