Metod för automatiserad sammanfattning och nyckelordsgenerering

Method for automated summary and keyword generator

Dennis Björkvall och Martin Ploug

Examensarbete inom
Datorteknik,
Grundnivå, 15 hp
Handledare på KTH: Jonas Wåhslén
Examinator: Ibrahim Orhan
TRITA-STH 2016:3

KTH
Skolan för Teknik och Hälsa
136 40 Handen, Sverige

Sammanfattning

Företaget Widespace hanterar hundratals ärenden i veckan vilket kräver stor överblick för varje anställd att sätta sig in i varje enskilt ärende. På grund av denna kvantitet blir uppgiften att skapa överblicken ett stort problem. För att lösa detta problem krävs en mer konsekvent användning av metadata och därför har en litteraturstudie om metadata, automatiserad sammanfattning och nyckelordsgenerering utförts.

Arbetet gick ut på att utveckla en prototyp som automatisk kan generera en sammanfattning av texten från ett ärende, samt generera en lista av nyckelord och ge en indikation om vilket språk texten är skriven i. Det ingick också i arbetet att göra en undersökning av tidigare arbeten för att se vilka system och metoder som kan användas för att lösa denna uppgift. Två egenutvecklade prototyper, MkOne och MkTwo, jämfördes med varandra och utvärderades därefter. Metoderna som använts bygger på både statistiska och lingvistiska processer. En analys av resultaten gjordes och visade att prototypen MkOne levererade bäst resultat för sammanfattningen och att nyckelordlistan tillhandahöll nyckelord av hög precision och en bred täckning.

Nyckelord

Metadata, sammanfattning, nyckelord, autogenerering, textanalysering

Abstract

The company Widespace handles hundreds of tasks (tickets) per week, which requires great overview by each employee. Because of this quantity, creating a clear view becomes a major problem. To solve this problem, a more consistent use of metadata is required, therefore, a study of metadata, automated summary and key words generation has been performed.

The task was to develop a prototype that can automatically generate a summary, a list of keywords and give an indication of what language the text is written in. It was also included in the work to make a survey of earlier works to see which systems and methods that can be used for this task. Two prototypes were developed, compared with each other and evaluated. The methods used were based on both statistical and linguistic processes. Analysis of the results was done and showed that the prototype MkOne delivered the best results for the summary. The keyword list contained many precise keywords with high precision and a wide coverage.

Keywords

Metadata, summary, keywords, auto generator, text analyzing

Förord

Denna rapport är ett resultat av ett examensarbete inom datateknik på Kungliga Tekniska Högskolan på uppdrag av företaget Widespace.

Under examensarbetets gång har vi från skolan haft handledaren Jonas Wåhslén som vi vill tacka för den hjälp vi fått.

Vi vill även tacka Glim Södermark på Widespace som har varit hjälpsam och inspirerande under arbetets gång.

Ordlista

API – en uppsättning regler som beskriver hur programvara kan interagera med varandra

Artificial Intelligence (AI) – vetenskapen om att bygga intelligenta system

Lingvistik – vetenskapen om mänskliga språk

Query – en begäran efter specifik information

Stokastisk process – en tidsordnad slumpprocess

Tokenisering – att dela upp en text i ord, fraser, etc

Innehållsförteckning

1 Inledning

1.1 Problemformulering

För företaget Widespace har det länge varit ett problem att hitta ett system för hantering av support-ärenden som möter deras krav. I dagsläget delar alla avdelningar en och samma inkorg för inkommande support-ärenden. Dessa ärenden sorteras sedan ut till respektive avdelning, samt anställd via en manuell process. Denna process sker via en avläsning av ärendet och en evaluering görs för att lägga ärendet på bäst ägnad avdelning och person.

Idén är att skapa ett system som kan innehålla flera inkorgar, som fördelas över flera avdelningar. Första sorteringen sker då redan när kunden initierar ett ärende. För att underlätta nästa steg i tilldelningsprocessen måste bättre metadata skapas för att möjliggöra för en anställd att bättre tyda och förstå ärendet. Med en sortering och automatisering av sammanfattande metadata kan Widespace undanröja flera tidskrävande processer och spara resurser.

1.2 Målsättning

Målet med detta projekt var att utveckla en webbapplikation till företaget Widespace för hantering och skapande av ärenden. Huvudmålet var att skapa en metod som automatiskt kan generera relevant metadata för skapade ärenden och detta för att underlätta tilldelning av ärenden till anställa. Metoden kommer att ta en text som kommer via mejl från en kund och utifrån detta skapa metadata som ska ge en sammanfattande beskrivning av ärendet.

Denna uppgift delades upp i följande delmål:

1. Dokumentera ämnet metadata
2. Undersöka befintliga metoder som kan användas för skapande av metadata
3. Välja lämpliga metoder
4. Utveckla prototyp
5. Utföra simuleringsfall
6. Mäta och validera prototypens effektivitet
7. Analysera och diskutera prototypens resultat

1.3 Avgränsningar

Widespace hanterar ärenden på flera olika språk, för detta krävdes en prototyp som kan hantera detta. En begränsning på sex språk gjordes och detta berodde på att metoderna som byggde de egenutvecklade prototyperna MkOne och MkTwo var från bland annat OpenNLP som bara hade tränade modeller för språken engelska, svenska, tyska, portugisiska, holländska och danska. Huvudmålet var att hantera texter skrivna på engelska eftersom den största delen av deras ärenden är skrivna i det språket.

Under utveckling av metoderna för sammanfattning av text visade det sig att tillgång till träningsdata, det vill säga gamla ärenden blev svår. En exportering var inte möjlig i deras nuvarande system och gav inte möjlighet att träna på data som prototypen skulle användas för i framtidigt bruk. En överenskommelse om att prototypen skulle testas på texter från Wikipedia gjordes.

För att hålla en kort och konkret sammanfattning gjordes en begränsning på 30 % av textens längd.

1.4 Författarnas bidrag till examensarbetet

Denna kandidatsavhandling har genomförts som ett samarbete mellan Dennis Björkvall och Martin Ploug. Martin har fokuserat på delen som genererar sammanfattning av text, medan Dennis har fokuserat på extrahering av nyckelord.

2 Teori och bakgrund

I detta kapitel kommer en genomgång av vilka ämnen som projektet kommer täcka i processen att automatiskt skapa metadata från en ostrukturerad text. Ett flertal metoder används för att göra detta möjligt.

Natural Language Processing (NLP) (se 2.1 Natural Language Processing) är huvudsakligen det viktigaste steget i processen, för att analysera och lära en dator vad text betyder måste flera mindre processer jobba tillsammans. Dessa processer hör under NLP och kommer tillsammans ge struktur i texten och göra det möjligt för andra verktyg att analysera den.

Med en strukturerad text att arbeta vidare med måste en sammanställning och analys av data göras i en form av textutvinning (se 2.2 Textutvinning). Syftet med detta är att hämta ut den information från texten som anses vara relevant.

Metadata kan kort och enkelt definieras som data om data (se 2.3 Metadata). Det vill säga data som på något vis beskriver annan data. Metadata kan även användas för att beskriva och lokalisera resurser och kan genom att använda en utav de många metadata-standarder som finns göras tillgängliga för andra system att hitta och använda.

För att strukturingen och analyseringen av text ska leverera rätt data måste ett förfaringssätt användas (se 2.4 Algoritmer och förfaranden), hur ska texten behandlas för att få fram rätt resultat varje gång. För strukturingen används metoder som har tränats upp med hjälp av algoritmer för att göra dom så smarta som möjligt. Analyseringen använder sig av algoritmer och förfarande för att hämta ut den information som anses vara relevant, samt möjlighet för att mäta hur väl information extraheras.

2.1 Natural Language Processing

Natural Language Proccessing (NLP) refererar till datorsystem som analyserar det mänskliga språket. Indata till ett sådant system kan bestå av text och tal och James F. Allen nämner i artikeln "*Encyclopedia of Computer Science: Natural Language Processing*" [2] flera exempel på syften med detta så som översättning av ett språk till ett annat eller summering av text.

Ett vanligt sätt att använda *Natural Language Processing* är genom att dela upp processen i flera steg. Varje steg är till för att urskilja den lingvistiska skillnaden för syntax, semantik och pragmatik. Först och främst kommer varje rad i texten analyseras i term av dess syntax, det vill säga dela upp meningar i olika satsdelar. Detta är till för att få en ordning och skapa struktur. I nästa steg undersöks och bestäms ordets betydelse och tolkas därefter. Till slut analyseras textens pragmatiska upplägg, vilken betydelse orden har i förhållande till varandra [3].

2.1.1 Standard NLP uppgifter

För att få maskiner att förstå det mänskliga språket finns olika tillvägagångssätt och uppgifter som NLP använder sig av. Collobert, Weston, Bottou, Karlen, Kavukcuoglu och Kuksa nämner i artikeln "*Natural Language Processing (Almost) from scratch*" [4] fyra standarduppgifter:

- **Part of speech tagging**
  Part of speech tagging (POS) uppgift är att markera varje ord med en unik tagg som indikerar dess syntaktiska roll, till exempel plural, substantiv eller adverb.

- **Chunking**
  Chunking innebär att dela upp texten i fraser på så vis att syntaktiska relaterade ord hamnar i samma fras. Dessa fraser kan inte överlappa varandra, det vill säga att ett ord bara kan finnas inom en och samma fras. Exempel på dessa fraser kan vara substantiv-fraser (SF) eller verb-fraser (VF). Inom en fras kommer varje ord att få en unik tagg tillsammans med dess fras-tagg, antingen en starttagg (S) som indikera vart frasen börjar eller en tagg som indikerar att ordet tillhör samma fras (T) [5].

- **Named Entity Recognition**
  Named Entity Recogniction (NER) identifierar specifika ord eller fraser (entiteter) och kategoriserar dom, till exempel en geografisk position, sjukdom, organisation eller ett namn. Precis som för chunking kommer NER att indikera var entiteten startar och vad som tillhör den [4].

- **Semantic Role Labeling**
  Semantic Role Labeling (SRL) innebär uppteckning av semantiska argument associerade med ett predikat eller verb av meningen och klassificera deras specifika roll i meningen. Ett exempel är “Martin köpte en glass till Dennis” verbet i denna mening är “att köpa” och används för att upptäcka predikatet *Martin* som representerar meningens säljare (agent), glassen representerar det som köptes (temat) och Dennis representerar mottagaren [6].

Precis som människor får lära sig att läsa texter och urskilja information, kräver också NLP algoritmer en lärandefas. Resultatet beror ofta på hur mycket träning algoritmerna får. Denna lärandefas innebär mängder av träningsmaterial som resulterar i att den i framtiden vet hur den ska processa och bearbeta liknande information. För att denna process inte ska återupptas varje gång, finns det redan tränade modeller tillgängliga.

2.1.2 Identifiera ett språk

Identifiering av språk är en av dom mest grundläggande stegen för system som ska analysera text och tal. För att en korrekt analys ska kunna ske måste systemet veta vilket språk som ska processas. Det finns flera tillvägagångsätt för att ta reda på detta och i artikeln “*Comparing methods for language identification*” [7] nämner Muntsa Padró och Lluís Padró att det mest vanliga är att använda sig av den lingvistiska information som finns, det vill säga speciella karaktärer eller karakteristiska sekvenser av bokstäver. Ett annat sätt att lösa detta är med hjälp av statistiska metoder. Exempel på en sådan metod är att jämföra texten med ett lexikon av ett känt språk och på detta sätt försöka hitta likheter. Apache Tika (se 3.3 Språkidentifiering) är ett ramverk som kan identifiera 18 olika språk genom att använda sig av en n-gram-algoritm (se 2.4.2 N-gram) för att identifiera språk.

2.1.3 Lemmatisering

Lemmatisering handlar om att sammanföra olika böjningar av ett ord, det vill säga återfå ett ord till dess grundform. Önskvärt vid lemmatisering är exempelvis att orden ”skriver” och ”skrev” ska kunna spåras tillbaks till sin grundform ”skriva”.

Lemmatisering kan vara användbart bl.a. vid indexering och sökningar. Detta är därför vanligt förekommande vid informationshämtning (se 2.2.1 Informationshämtning). Detta för att det vid sökning efter ett ord ofta är relevant att visa resultat som innehåller sökordet i andra former.

Själva lemmatiseringen av ett ord kan genomföras med hjälp av flertalet olika typer av metoder. Dessa metoder varierar i tillvägagångssätt och kvalitet på resultat. En vanligt förekommande och relativt simpel metod är att analysera ordets suffix och genom att följa vissa grammatiska regler gå tillbaks till grundform. Träffsäkerheten för denna metod kan ökas genom att använda sig utav ordets POS-tagg för att identifiera vilka grammatiska regler som ska användas. Reglerna för hur ett ord böjs kan bero på dess ordklass. Även n-gram (se 2.4.2 N-gram) används i vissa lemmatiserings-metoder.

2.2 Textutvinning

Textutvinning eller ”*Text mining*”, också kallad “*Text Data Mining*”, är ett sätt att hämta information från data, med data menas samlingar av text. Eftersom att text inte är numeriskt mätbar krävs det en process för att kunna tyda, det är inte möjligt att söka efter specifika saker i data om inte struktur finns. I boken “*The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*” [8] förklaras flera tekniker som textutvinning använder sig av:

- Informationshämtning
- Datautvinning
- Informationsextraktion
- Maskininlärning

2.2.1 Informationshämtning

Som en del av textutvinning måste data erhållas för att sedan kunna användas. Informationshämtning, “*information retrieval*” eller ”IR”, är en teknik som används för att hitta relevant information från en större kollektion av data. Sökningar kommer i form av “*queries*” som kan baseras på metadata (se 2.3 Metadata), taggning eller en annan form av indexering. Exempel på informationssatsningssystem är sökmotorer som utifrån sökkriterier försöker hitta det som är mest relevant mot din sökning, informationen presenteras i dess relevanta ordning [9].

En tagg är ett icke-hierarkiskt nyckelord eller term tilldelad till data (bild, text, fil osv). Denna form av metadata hjälper att beskriva föremålet och tillåter data att bli hittad igen utifrån taggen. För IR kan taggning användas som sökkriterier. I NLP förekommer taggning i flera olika processer så som *Named Entity Recognition* vart taggning används för att markera vilken kategori som ett eller flera ord tillhör. *Part of Speech* använder sig av taggning för att markera ett ords semantiska roll i en fras.

2.2.2 Datautvinning

Datautvinning (*data mining*) är ett verktyg för att söka efter mönster, samband och trender i stora data mängder. Verktygen använder statistiska beräkningsmetoder kombinerat med algoritmer för till exempel maskinlärning (se 2.2.4 Maskinlärning) eller mönsterigenkänning. Ett av kraven för att datautvinning ska appliceras direkt på datasamlingen är att det måste finnas någon form av struktur, exempel på strukturerad data är en relationsdatabas. För att analysera ostrukturerad data, i detta fall en textmassa krävs andra verktyg för att kunna tyda informationen innan en datautvinning kan gå till. NLP (se 2.1 Natural Language Processing) och informationsextraktion (se 2.2.3 Informationsextraktion) är båda metoder för att ge en ostrukturerad datasamling struktur.

2.2.3 Informationsextraktion

Informationsextraktion (IE) används för att hitta relationer i ostrukturerad information och ge det struktur. Ett exempel kan vara en stor samling av text som innehåller viktig information, för att få tillgång till detta skulle det vara simpelt och lätt om en enkel "*querie*" kunde ställas för att erhålla detta. Men eftersom att text är ostrukturerad är detta inte möjligt, något måste göras för att ge texten struktur. Detta är den huvudsakliga uppgiften för IE, att skapa struktur i text. De data som IE genererar är av tydlig struktur som kan användas för att få fram relevant information. Denna process kan göras på flera olika sätt, några av dom viktigaste stegen är att hitta entiteter (se 2.1 Natural Language Processing) och hitta relationer mellan entiteterna.

2.2.4 Maskinlärning

Maskininlärning, "Machine learning", är en typ av artificiell intelligens (AI) som ger datorer möjlighet att lära sig. Maskininlärning fokuserar på utveckling av datorprogram som kan lära sig att växa och förändras när de utsätts för ny data. I boken "*Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*" [10] nämns hur machine learning kan användas för att hitta mönster i sammanhang med datautvinning och utifrån tidigare data förutspå framtida data.

2.3 Metadata

Metadata är strukturerad data som beskriver, förklarar eller lokaliserar en resurs, vanligtvis i form av en fil, och på så vis förenklar vidareanvändning av resursen [11]. Med strukturerad data menas i detta fall att beskrivning av resursen sker med hjälp av attribut. Metadata kan antingen sparas som en del av resursen eller externt i en databas. Fördelen med att spara metadata som en del av resursen är att den beskrivande informationen uppdateras tillsammans med dess data men försvårar dock skrivning av ny data i filen. Genom att spara metadata i en databas kan sökningar efter specifika resurser utföras mer effektivt [11].

Metadata kan generellt delas upp tre olika kategorier [11,12]:

- **Beskrivande metadata** beskriver en resurs innehåll med hjälp av element som definierar titel, författare, etc.
- **Strukturell metadata** beskriver hur innehållet i en resurs är organiserat.
- **Administrativ metadata** beskriver själva resursen med attribut som datum när resursen skapades och av vilken typ resursen är. Beskrivningen kan även innehålla information om resursens rättigheter och restriktioner. Denna kategori kan ibland delas upp i de två delarna *bevarande metadata* och *rättighetshanterade metadata*.

2.3.1 Standarder inom metadata

Syftet med att använda sig utav metadata är att beskriva information på ett sätt som både människor och maskiner enkelt kan förstå. Det finns därför många standarder etablerade för att underlätta beskrivning av olika typer av data. Varje enskild standard innehåller ett metadata-schema, en struktur bestående av element, som måste följas för att på ett smidigt sätt kunna dela information av samma typ. De flesta av dessa scheman har syntaxer som i sin tur följer standarder som SGML (*Standard Generalized Mark-up Language*) och XML (*Extensible Mark-up Language*) [12]. I följande avsnitt beskrivs de metadatastandarder som för projektet anses vara relevanta, det vill säga de standarder som stödjer text i digitalt format.

2.3.2 Dublin Core

Dublin Core är en av de mest använda metadatastandarderna på internet [13]. Standarden togs fram 1995 och bestod då av 15 element (attribut) till för beskrivning av en resurs (se Tabell 1). Dessa element kallas *Dublin Core Metadata Element Set* (DCMES) och är uppdelade i tre kategorier [14]. Antalet element har sedan dess uppdaterats till att idag innehålla betydligt många fler. Standarden är konstruerad för att kunna stödja alla typer av resurser, både abstrakta resurser, som filer, och fysiska objekt.

Tabell 1: Dublin Core Metadata Element Set (Uppdelade i kategorier)

Content	*Intellectual Property*	*Instantiation*
Title	Creator	Date
Subject	Publisher	Format
Description	Contributor	Identifier
Type	Rights	Language
Source		
Relation		
Coverage		

2.3.3 The Text Encoding Initiative

The Text Encoding Initiative (TEI) är ett metadataschema framtaget för att underlätta strukturering och utbyte av textmassor. Standarden innehåller ett flertal notationer för att dela upp text i stycken, meningar, namn, etc. [15,16]. Schemat följer standarden SGML.

De grundläggande notationerna i TEI vid namn *Core tagset [15]:*

- Paragrafer
- Segmentering, exempelvis i form av ortografiska meningar
- Listor
- Markering av fraser (highlight)
- Citering
- Namn, nummer, enheter, datum och tid, och liknande typer av data
- Grundläggande ändringar, så som fel, borttagningar och tillägg
- Länkar och korsreferenser
- Existerande notationer och indexeringar
- Uppmärkning av verser, etc.
- Bibliografiska citeringar

2.3.4 Metadata Encoding and Transmission Standard

Metadata Encoding and Transmission Standard (METS) är en metadatastandard under utveckling på begäran av *Digital Library Federation*. Standarden är till för att beskriva objekt i digitala bibliotek vilket görs i form av ett METS-dokument, enligt XML, bestående av sju sektioner [17].

De sju sektionerna, tillsammans med dess tagg-namn, i ett METS-dokument förklaras nedan [18]:

- METS-header (metsHdr)
- Beskrivande metadata (dmdSec)
- Administrativ metadata (amdSec)

- Filsektion (fileSec)
- Strukturschema (structMap)
- Strukturlänkar (structLink)
- Beteende (behaviorSec)

Enbart första och femte sektionen, METS-*header* och *strukturschema*, är obligatoriska att använda. Sektionerna *beskrivning*, *administrativt* och *beteende* kan väljas att placeras externt [19].

2.3.5 Metadata Object Description Schema

Metadata Object Description Schema (MODS) är en XML-baserad standard lämpad för bibliografi som har skapats i syfte att vara ett svar på ett simplare alternativ till standarden MARC genom att använda mer användarvänliga taggar som lättare kan läsas av människor [19, 20].

2.4 Algoritmer och förfaranden

I detta avsnitt presenteras ett antal olika algoritmer och förfaranden som är relevanta för projektet. Dessa algoritmer och förfaranden är relevanta för olika delar och funktioner av det system som utvecklats, däribland NLP och viktning av text.

2.4.1 Dold Markovmodell

En dold Markovmodell kan ses som en samling tillstånd som ett system rör sig emellan. Dessa tillstånd förhåller sig till varandra enligt deras sannolikhetsfördelning över möjliga utfall. Algoritmen är bestående av en dubbelstokastisk process, med en underliggande stokastisk process. Den underliggande stokastiska processen är inte observerbar, utan kan endast observeras genom symboler som emitteras i tillstånden [21, 22]. Algoritmen är implementerad och används inom många olika områden, bl.a. i NLP-uppgifter som *POS* och *chunking* [23].

2.4.2 N-gram

N-gram används ofta vid beräkningar som gäller lingvistik och sannolikhet, så som att förutspå ett nästa objekt. Ett n-gram innehåller en teckensekvens med n antal tecken från en sträng. Ett n-gram med längden; ett tecken kallas uni-gram, två tecken kallas bi-gram, tre tecken kallas tri-gram, och därefter kallas de fyra-gram, osv.

2.4.3 Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TF-IDF) är en numerisk statistisk metod vars syfte är att beräkna vikten av en term (t) i ett dokument (d) som en är del av en korpus (D). Som kan ses i ekvation 3 är TF-IDF en produkt av de två statistiska metoderna *termfrekvens (tf)* och *inverterad dokumentfrekvens (idf)*.

$$tf(t,d) = f(t,d) \tag{1}$$

Termfrekvensen används till att beräkna vikten av en term i ett dokument. Detta görs vanligtvis genom att räkna hur många gånger termen förekommer i dokumentet, då upprepning av termen är positivt för dess viktning [24]. Det finns även andra sätt att definiera termfrekvensen, bl.a. genom att använda sig av en logaritmisk eller normaliserad viktskala. Ekvation 1 visar ett grundläggande sätt att räkna ut termfrekvensen [25], som erhålls i form av ett heltal mellan 0 och antalet termer i dokumentet.

$$idf(t, D) = log\left(\frac{N}{n}\right) \quad (2)$$

Inverterad dokumentfrekvens används till att vikta en term i en korpus. Detta görs genom att räkna hur många dokument i en korpus som innehåller termen. Viktningen fungerar tvärtom mot termfrekvens, termens vikt påverkas negativt om den finns i många dokument, då termen inte anses vara lika beskrivande. Ekvation (**2** visar ett grundläggande sätt att räkna ut idf [25], där *n* är antalet dokument termen finns i och *N* är antalet dokument i korpusen.

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) \quad (3)$$

2.4.4 Precision och Täckning

För att kunna analysera data från ett resultat måste det finnas ett sätt att värdera kvaliteten av resultatet. För detta krävs testdata som innehåller två typer; relativ data och irrelevant data. Syftet med detta är att kunna tyda mellan vad som är rätt och vad som är fel. Två utvärderingsmått används, täckning och precision [26]. Täckning anger hur stor andel av de relevanta data som finns i resultatet. Precision anger hur mycket utav resultatet som är relevant. För att få en bild av kvaliteten för resultatet används ett F värde, i artikeln "*The truth of f-measure*" [27] definieras F värdet som ett genomsnitt av täckning och precision.

$$Precision = \frac{relevant\ data \cap resultat\ data}{resultat\ data} \quad (4)$$

$$Täckning = \frac{relevant\ data\ \cap resultat\ data}{relevant\ data} \quad (5)$$

$$F = 2\frac{Precision \times Täckning}{Precision + Täckning} \quad (6)$$

F-modellen är ett statistiskt verktyg för att testa träffsäkerhet. Det bästa värdet i F-modellen är 1,0 som betyder att träffsäkerheten är 100 % och det sämsta värdet är 0,0 med en träffsäkerhet på 0 %.

2.4.5 Automatiserad Summering

En sammanfattning, är en kortare summering av en längre text som innehåller det viktigaste från texten. För att evaluera vad som anses vara viktigast i en text kan olika tillvägagångs sätt användas.

I arbetet "*Email Classification and summarization: A machine learning Approach*" [28] presenterades en algoritm som är byggd på en statistisk process som ser upp för mest frekventa ord i en text och dess relation till varje rad. En ordning av alla rader görs efter deras innehåll av frekventa ord genom texten och översta raderna kommer att användas som sammanfattning.

Rushdi Sahms, M.M.A. Hashem, Afrina Hossain, Suraiya Rumana Akter och Monika Gope har i sin rapport "*Corpus-based Web Document Summarization using Statistical and Linguistic Approach*" [29] observerat hur man kan förbättra en sammanfattning genom att använda statistiska och lingvistiska processer med fokus på subjektiv. Rapporten har framför allt studerat innebörden av upprepade termer och hur meningar i en text kan listas beroende på dess relevans.

För att mäta hur precis en sammanfattning är kan vara svårt, det finns inte direkt ett rätt svar. I arbetet "*Email Classification and summarization: A machine learning Approach*" [28] presenterades

en algoritm för summering av innehållet i e-post. Som referens för jämnföring av deras algoritm användes en mänsklig summering. Den mänskliga summeringen innebar att flera personer skulle välja ut dom meningarna i texten som ansågs vara bäst lämpad för att summera texten.

För att skapa en sammanfattning finns två olika tillvägagångs sätt [30]:

- Extrahering, fungerar på så sätt att ett urval av ord eller fraser väljas ut ur texten som då blir summeringen av texten.
- Abstrahering, detta sätt är till för att skapa en mer människolik summering. Med användning av NLP tekniker kan en summering göras med hjälp av det lingvistiska innehållet av texten.

3 Metoder och resultat

Detta kapitel beskriver de metoder och verktyg som användes för att utveckla den prototyp som generar relevant metadata och hur dom kommer att användas (se 3.1 Tolka användarbeteende). För att få en förståelse för ämnet genomfördes en litteraturstudie med syfte att undersöka och dokumentera ämnet metadata, vilka typer av metadata som finns samt vad som anses vara relevant metadata i projektets sammanhang (se 3.2 Relevanta metadata).

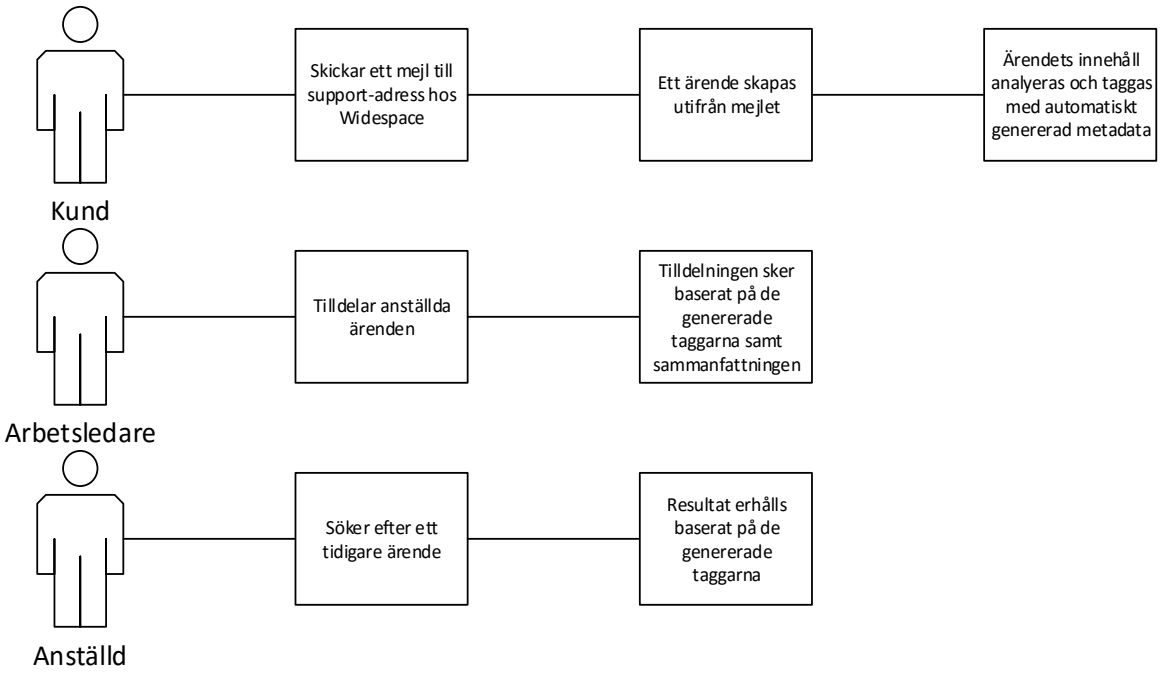
För att generera bästa möjliga data genomfördes en litteraturstudie och undersökning med befintliga metoder som kan användas för skapande av metadata. En av huvudstegen för användning av dessa metoder är att kunna identifiera språket (se 3.3 Språkidentifiering). Med en språkidentifiering kan rätt metoder användas för att ge texten struktur (se 3.4 Strukturering). Efter strukturering av texten är det sedan möjligt att börja fokusera på vad som är relevant (se 3.5 Text av högre kvalitet). För bäst möjliga redovisning av relevant data tillämpade två sammanfattnings metoder och en metod för generering av nyckelord (se 3.6 Strategi för sammanfattning och nyckelord). Utifrån denna undersökning gjordes en kvalitativ bedömning av vilka metoder som är bäst lämpade (se 3.7 Modell).

Med hjälp av utvalda metoder skapades två prototyper, MkOne och MkTwo (se 3.8 Prototyper). Syftet med dessa prototyper var att testa deras effektivitet i förhållande till människan och varandra (se 3.9 Test och validering). Resultaten jämfördes och redovisades i form av grafer (se 3.10 Sammanställning av resultat).

3.1 Tolka användarbeteende

Det finns tre aktörer som är tänkta att vara en del av den utvecklade metoden (se Figur 1). Dessa tre aktörer är *Kund, Arbetsledare* och *Anställd*. Kund är den aktör som inleder ett ärende. Detta går till genom att kunden skickar en e-post till någon av företagets supportadresser. Det är utifrån innehållet i denna e-post som metadata genereras.

Arbetsledare refererar till den aktör som har ansvaret att tilldela inkomna ärenden till andra anställda. Själva bedömningen av vilken anställd som är bäst lämpad att ansvara för ärendet kan snabbbare och lättare bestämmas med hjälp av de genererade nyckelorden och sammanfattningen. Anställd är den aktör som blir tilldelad ansvaret att följa upp ett ärende skapat utifrån en e-post från Kund.



Figur 1: Händelseschema för de olika aktörerna

3.2 Relevanta metadata

Grundläggande för att skapa metoden hämtades information om vad som anses vara relevant metadata, eftersom vad som anses vara relevant är relativt och baseras på vem som ska använda metoden. Urvalet gjordes tillsammans med Widespace. Inom Widespace finns det olika avdelningar som alla kommer ta användning av metoden, för detta krävdes en bredare bild av vad som anses relativt.

- **Språk**, det språk som ett ärende skrivs i. “My name is Einstein, I’m 120 years old” kommer att identifieras som engelska.
- **Sammanfattning av innehåll**, en kort sammanfattande text om innehållet i ett ärende. I form av text och nyckelord.
- **Namn**, alla namn så som Albert Einstein, Martin Ploug och Dennis Björkvall.
- **Företag**, alla företagsnamn så som IBM, Apple och Microsoft.

3.3 Språkidentifiering

Identifiering av språk är en av de mest grundläggande stegen för system som ska analysera text och tal. För att en korrekt analys ska vara möjlig måste systemet veta vilket språk som ska processeras. Det finns flera tillvägagångsätt för att ta reda på detta, mest vanliga är att använda sig av den lingvistiska information som finns, d.v.s. speciella karaktärer eller karakteristiska sekvenser av bokstäver [8]. Ett annat sätt att lösa detta på är med hjälp av statistiska metoder. Ett exempel på en statistisk metod är att jämföra texten med ett lexikon för ett känt språk och på så sätt försöka hitta en likhet. För språkidentifiering i prototypen används API’et Apache Tika.

3.4 Strukturering

För att kunna utvinna relevant data från en textmassa krävs det att texten struktureras upp, detta är ett av grundstegen i informationsextraktion (se 2.2.1 Informationshämtning). För att göra detta möjligt användes OpenNLP, som med hjälp av färdigtränade modeller gjorde det möjligt att strukturera upp texten på sex olika språk. Strukturingen sker i två steg, först delas upp texten i meningar. Detta görs med hjälp av metoden *sentence detection*. En mening identifieras och läggs i en matris. Varje

mening kommer sedan att behandlas med hjälp av metoden tokenisering som kommer dela upp varje ord i en 1x1 matris. Struktureringen kommer att resultera i en 1x1 matris som innehåller flera 1x1 matriser. Efter att varje mening och ord har identifierat i texten kan texten behandlas efter data som har högre kvalitet, relevans.

3.5 Text av högre kvalitet

Eftersom att en text innehåller mycket data som anses vara irrelevant för metoden kan en bortsortering av dessa data göras. För att ta reda på vad som anses vara irrelevant används lingvistiska och statistiska processer. En stoppordslista används för att ta bort ord som inte har någon högre betydelse för textens innehåll, detta är en statistisk process som tar bort ord så som "och", "eller", "men", "en", "ett" och "att". För metoden användes färdiga stoppordslistor. Metoden kan hantera flera olika språk och krävde en stoppordslista för respektive språk. För projektet användes redan tränade stoppordlistor.

För att hitta ordbasen för ett ord användes lemmatisering (se 2.1.3 Lemmatisering) i metoden. Syftet med detta är att undvika att ord upprepas i texten i form av annan böjning. Ett exempel är en elefant, flera elefanter. Ordet kommer efter en lemmatisering att finnas bara i dess basform, det vill säga elefant. Detta görs för att öka ordets relevans i texten, om ett ord upprepepas oftare får det en högre betydelse för texten.

3.6 Strategi för sammanfattning och nyckelord

En sammanfattning, ofta även benämnd *abstract*, är en kortare summering av en längre akademisk text. Dess längd kan variera beroende på totala textens längd, för metoden kommer den längden att bestå av runt 30 % av textens längd. Sammanfattningen består av dom mest relevanta raderna i en text. Syftet är att ge den tänkbara läsaren en uppfattning om texten i sin helhet utifrån en mindre text baserat på den fullständiga texten.

För båda sammanfattningsmetoderna studerade under litteraturstudien (se 2.4.5 Automatiserad Summering) bygger sammanfattningen huvudsakligen på upprepade termer i en text. Dessa termer kommer att anses som nyckelord för sammanfattningen och kommer även presenteras för användaren som en mindre sammanfattning. Eftersom att kvantiteten av texten kan variera mycket måste nyckelord också hittas på andra sätt utöver frekventa termer. För detta användes Named Entity Recognition (se 2.1.1 Standard NLP uppgifter) för att hämta ut namn och företagsnamn.

3.7 Modell

Här beskrivs de metoder och ramverk som användes för att skapa den modell som tagits fram för att skapa automatiserad metadata (se Figur 2). Det användes tre stycken API'er för att utveckla metoden. Ett av de första kraven till valet av API'er var dess förmåga att hantera flera språk, först och främst engelska. API'erna som användes är OpenNLP (se 3.7.1 OpenNLP), Apache Tika (se 3.7.2 Apache Tika) och Tartarus Snowball (se 3.7.3 Tartarus Snowball). För mer information om hur dessa användes se avsnittet om den utvecklade metoden (se 3.8 Prototyper).

Eftersom att språket kan variera för varje text måste en identifiering göras, för detta användes Apache Tika. Apache Tika kan identifiera 18 språk och inkluderade dom språken som ansågs vara relevanta för metoden. Med en identifiering av språket som texten är skriven i kan nästa steg i processen förekomma. För att strukturera upp texten krävdes det ett API som innehåll färdigtränade modeller för sentence detection och tokenisering. OpenNLP innehåller detta, samt en färdig tränad modell för Named Entity Recognition (se 2.1.1 Standard NLP uppgifter) för engelsk text.

3.7.1 OpenNLP

OpenNLP är ett Java-bibliotek för naturliga språk (NLP), som utvecklats under Apache licens och är OpenSource vilket gör det möjligt att använda fritt. NLP som domän, behandlar interaktionen mellan datorer och det mänskliga språket. Det huvudsakliga målet i detta fall är att göra det möjligt för datorer att utvinna meningar ur naturligt språk. OpenNLP innehåller flera verktyg så som:

- Meningsuppdelning
- Tokenisering
- Named Entity Recognition

3.7.2 Apache Tika

Apache Tika är ett verktyg och ett projekt under Apache Software Foundation som kan användas till att extrahera strukturerad text och metadata från över tusen olika filtyper. Tika kan även identifiera texters språk med hjälp av både lingvistiska och statistiska metoder. Verktyget är licenserat enligt Apache licence och är tillgängligt i form av OpenSource. Apache Tika var tidigare en del av det välkända indexeringsramverket *Lucene*.

Från detta ramverk användes enbart ovannämnda funktion som gör det möjligt att identifiera språket för en text.

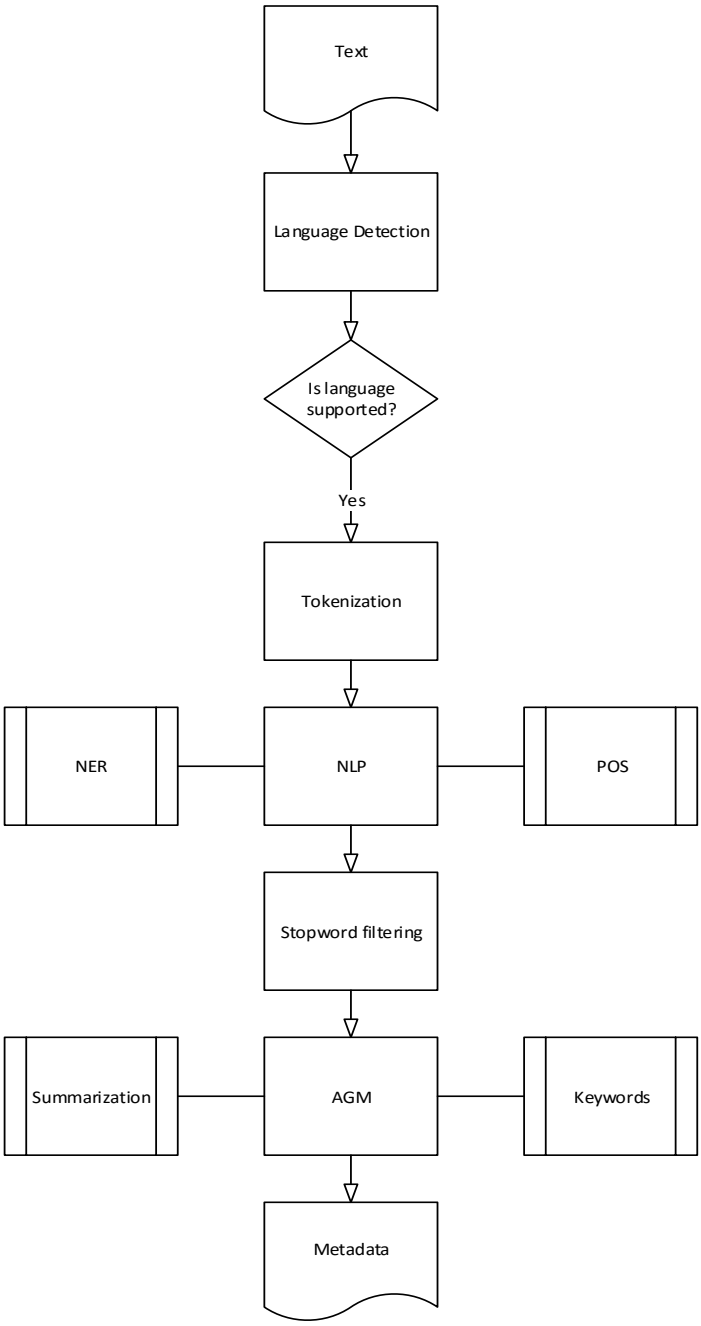
3.7.3 Tartarus Snowball

Tartarus Snowball är ett OpenSource bibliotek, enligt licensen BSD, som innehåller funktioner för att lemmatisera ord. Biblioteket är skrivet i scriptspråket *Snowball* som kan kompileras till körbar Javakod.

3.8 Prototyper

Utifrån den framtagna modellen (se 3.7 Modell) skapades två prototyper. Huvudmålet med MkOne var att skapa en kort sammanfattning och en lista av nyckelord utifrån en text. Fasen för sammanfattning gjordes på två olika sätt. Meningen med detta var att testa en metod som använder sig av bara statistiska processer och en metod som även använder sig av lingvistiska processer.

Modellen som togs fram som grund för prototyperna (se Figur 2) visar de metoder som texten kommer behandlas i enligt kronologisk ordning. För att utveckla prototyperna användes Apache Tika, OpenNLP och Tartarus Snowball.



Figur 2: Flödesschema som beskriver stegen i den egenutvecklade modellen

Första steget i figur 2 är språkidentifiering, metoden tar emot en text i form av en JSON-fil. Språkidentifieringsmetoden använder sig av Apache Tika och returnerar en sträng som innehåller landskoden i ISO 3166-format. Metoden stödjer sex olika språk (engelska, svenska, danska, holländska, tyska och portugisiska) och om ett annat språk identifieras kommer metoden inte att användas. Texten skickas vidare till steg två, identifiering av rader. Metoden tar emot texten och processera den utifrån dess språk, metoden är en förprocess till tokenisering och är tagen från OpenNLP. Syftet med metoden är att dela upp texten i rader (meningar), detta görs genom att ta reda på punkterna i texten, men även se upp för punkter som inte indikera slutet på en rad. Ett exempel på detta kan vara Mr. Ploug eller Mrs. Björkvall. Varje rad läggs i en matris och returneras.

Matrisen med rader skickas sedan vidare för tokenisering. Denna metod använder sig av färdig-tränade modeller från OpenNLP och bryter upp texten i ord, punkter, frågetecken m.m. Metoden sparar dessa så kallade tokens i en matris. Eftersom att varje rad måste tokeniseras kommer en matris av matriser att returneras.

Matrisen av matriser skickas vidare till metoden för ordklasstaggning (se 2.1.1 Standard NLP uppgifter). Denna metod togs från OpenNLP och använder sig av tränade modeller för språket som texten är skriven i. Ordklasstaggaren markerar upp alla tokens med dess respektive ordklass: Substantiv, verb, egennamn m.m. Metoden returnerar en matris med tokens respektive ordklass.

Nästa steg i prototyperna är att identifiera stoppord, för detta användes redan gjorda listor [31]. Metoden tar emot en matris av matriser innehållande tokens och jämför varje token mot listan, tokens som matchas kommer att tas bort från matriserna. Denna metod skapades på egenhand och använder sig inte av ett API.

För att få fram dom första nyckelorden ur texten används Named Entity Recognition (se 2.1.1 Standard NLP uppgifter). Metoden hämtas från OpenNLP och extraherar alla företagsnamn och namn från texten, dessa entiteter returneras i en matris som kommer framöver att nämnas som nyckelordlistan.

De första nyckelorden har tagits fram utifrån en tränad modell (se 2.1.1 Standard NLP uppgifter). Denna modell kan inte se upp för termer som är återkommande i texten. För att ta fram dessa nyckelord används en termfrekvensmetod, metodens syfte är att skapa en lista över termer som repeteras oftare än en gång i texten. För att metoden skulle ge bäst möjliga resultat krävdes det att termer som kan förekomma i andra former än grundform skulle elimineras. Detta gjordes med hjälp av lemmatisering, en metod från Tartarus Snowball.

Termfrekvensmetoden baseras på TF-delen i TF-IDF (se 2.4.3 Term Frequency - Inverse Document Frequency) och returnerar en matris med upprepande termer samt hur ofta dom upprepas. IDF-delen i TF-IDF implanterades som en del av metoden, men gav ingen skillnad då testerna gjordes på individuella dokument. Termer som upprepas mera än en gång kommer att läggas till nyckelordlistan.

Sista delen i prototyperna är att skapa en sammanfattning av texten. Det undersöktes två olika metoder för detta, ena metoden kommer att fortsätta under namnet MkOne och andra kommer att gå under namnet MkTwo. MkOne och MkTwo resulterar båda i en sammanfattning av text, en nyckelordlista och vilket språk som texten förekommer i. Det som skiljer prototyperna åt är sättet de genererar sammanfattning på, de kommer identifiera samma språk för texten och extrahera samma nyckelord.

3.8.1 Sammanfattning MkOne

Sammanfattningsmetoden för MkOne lägger störst fokus på upprepade termer i texten samt meningars position. Frekventa termer anses vara relevant för texten och rader som innehåller dessa termer kommer få större betydelse. En menings anses ha större relevans beroende på hur högt den placeras i texten. Sammanfattningen bygger på de meningarna med högst betydelse.

1. Vikta varje mening med dess position
2. Identifiera N mest frekventa ord i texten
3. Lista M meningar från texten som innehåller flest frekventa ord

4. Rangordna listan med meningarna beroende på hur många frekventa ord som meningen innehåller samt dess positionsvikt.

3.8.2 Sammanfattning MkTwo

Sammanfattningsmetoden för MkTwo lägger även fokus på det lingvistiska. Metoden kommer använda sig av en ordklasstagger för att ta reda på substantiv och subjekt för varje rad. Första delen går ut på att vikta varje mening utifrån hur många substantiv som den innehåller. Formeln för detta visas i ekvationen 7. För normalisering av vikten kan ekvation 8 användas.

$$t_n = Antalet\ substantiv\ i\ raden$$

$$w_n = Antalet\ ord\ i\ raden$$

$$tf_k = Term - frekvensen\ för\ substantivet$$

$$S_1 = \frac{t_n}{w_n} \times \sum_{K=1}^{t_n} tf_k \tag{7}$$

$$SW = \frac{S_1}{max(S_1)} \tag{8}$$

Andra delen av metoden är till för att ta reda på subjektet för varje rad och vikta raden ännu en gång. Syftet med detta är att få fram dom raderna som innehåller ett subjekt, ett ämne som omtalas och behöver framhävas. Formeln för detta visas i ekvation 9.

$$tv_n = Subjektet\ i\ meningen\ (substantivet\ innan\ verbet)$$

$$S_2 = \sum_{K=1}^{tv_n} tf_k \tag{9}$$

$$SuW = \frac{S_2}{max(S_2)} \tag{10}$$

Summeringen för dessa två delar blir vikten för raden. Vikterna rangordnas i en lista. 30 % av de tyngsta vikterna (raderna) kommer att vara sammanfattningen för texten.

$$R = SW + SuW \tag{11}$$

3.9 Test och validering

För att testa och validera metodens funktionalitet utfördes tester på 15 texter från artiklar tagna från Wikipedia (se Bilaga 1-2). Dessa texter valdes inte utifrån några specifika kriterier, utan det enda kravet var att texterna skulle innehålla 9-15 meningar. Testerna bestod av två delar. Den första delen av testet validerar funktionaliteten för extrahering av nyckelord. Den andra delen validerar funktionaliteten för sammanfattning. Texterna var skrivna på engelska och bestod av 9-15 meningar. Dessa texter kommer härefter refereras till som test-dokumenten. För testerna användes tio personer för validering av nyckelord och sammanfattning.

3.9.1 Validering av funktionalitet för extrahering av nyckelord

För att erhålla resultatet av testerna i ett mått som kan mätas och jämföras användes F-Modellen (se 2.4.4 Precision och Täckning) och dess två inre komponenter *Precision* och *Täckning*. Dessa värden erhölls genom att först märka upp texten i respektive test-dokument enligt de kategorier som ansågs vara relevanta (se 3.2 Relevanta metadata). Efter detta kördes metoden med de olika texterna som indata, resultatet (utdatat) observerades och de olika måtten beräknades för varje enskild text med uppmärkning och erhållet resultat som grund.

3.9.2 Validering av funktionalitet för sammanfattning

Även för testet som validerar funktionaliteten för sammanfattningen användes F-modellen för att erhålla ett mätbart resultat. Inför denna validering genomfördes en småskalig undersökning som användes som referens vid beräkning av F-värde. Undersökningen innebar att personerna fick läsa test-dokumenten och för varje text välja ut en tredjedel av meningarna som de tyckte sammanfattade texten bäst. De fick även rangordna de meningar som valdes utifrån hur viktiga de ansågs vara.

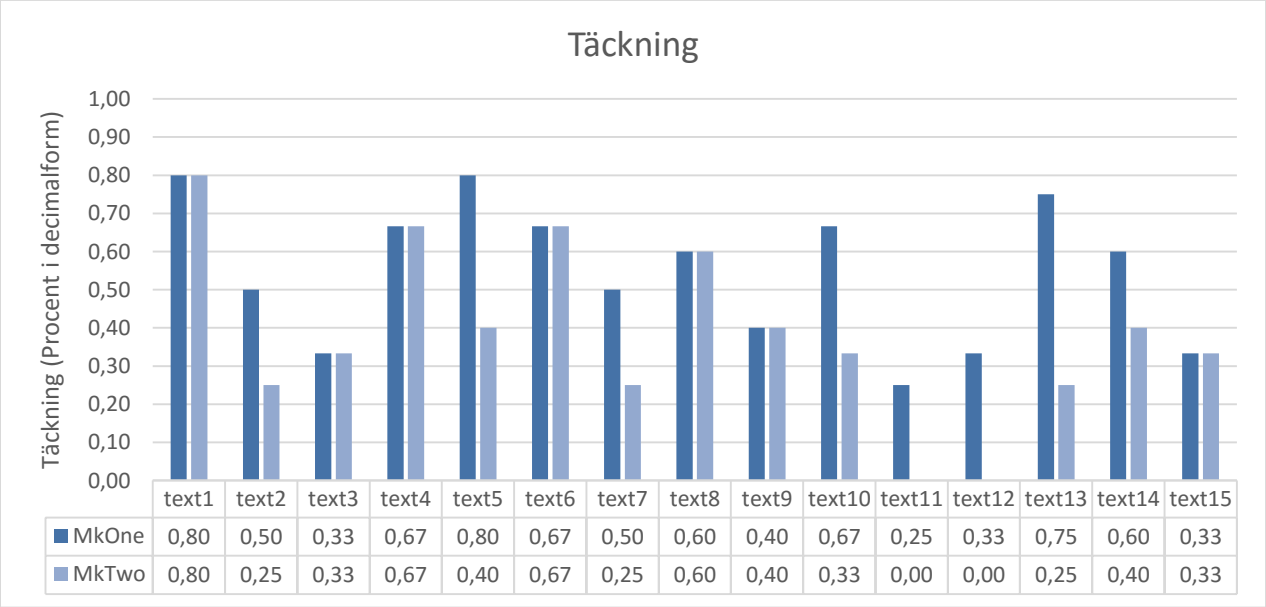
3.10 Sammanställning av resultat

I detta avsnitt följer en sammanställning av resultaten från testerna. Följande grafer indikerar täckning, precision och F-värde för sammanfattnings-metoder i prototyperna MkOne och MkTwo, samt prototypernas förmåga att hitta relevanta nyckelord i test-dokumenten.

Resultat som visualiserar prototypernas förmåga att identifiera textspråk presenteras inte i detta avsnitt då denna funktion erhöll ett resultat som innebar att språket i alla test-dokument identifierades korrekt.

3.10.1 Sammanställning för sammanfattningsmetoderna

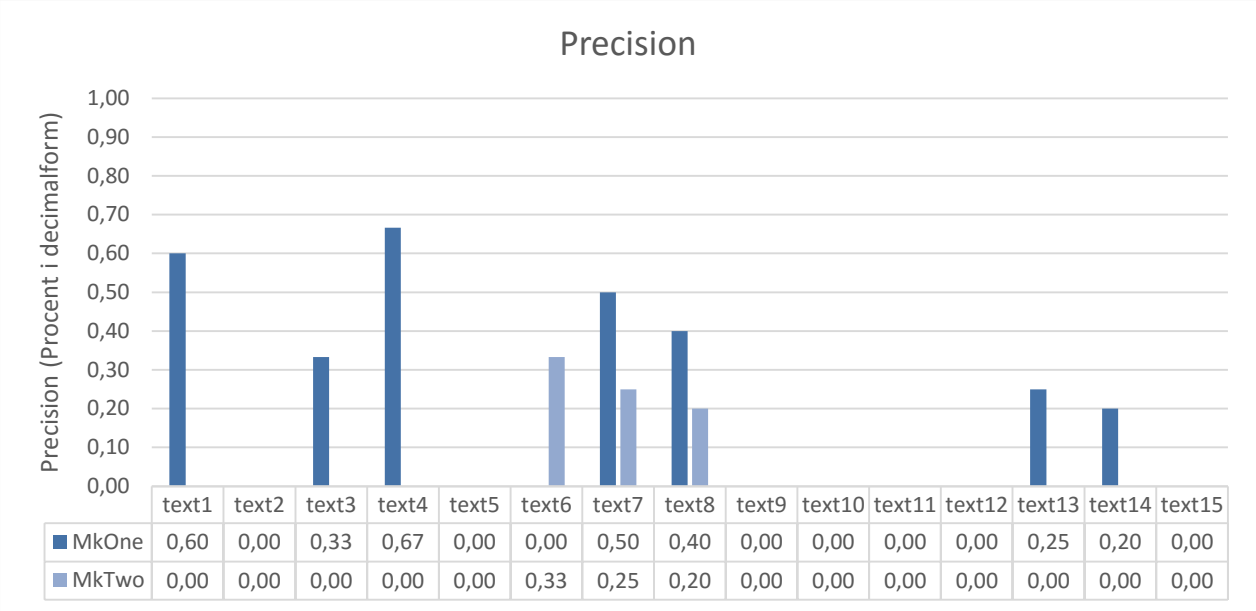
Den första figuren visar hur stor täckning de två metoderna har (se Figur 3). MkOne fick en genomsnittlig täckning på 55 %, det vill säga att lite över häften av alla fördefinierade meningarna hittades. MkTwo hade en täckning på 38 % (se Figur 6).



	text1	text2	text3	text4	text5	text6	text7	text8	text9	text10	text11	text12	text13	text14	text15
MkOne	0,80	0,50	0,33	0,67	0,80	0,67	0,50	0,60	0,40	0,67	0,25	0,33	0,75	0,60	0,33
MkTwo	0,80	0,25	0,33	0,67	0,40	0,67	0,25	0,60	0,40	0,33	0,00	0,00	0,25	0,40	0,33

Figur 3: Mätningar för de två prototypernas täckning på test-dokumenten

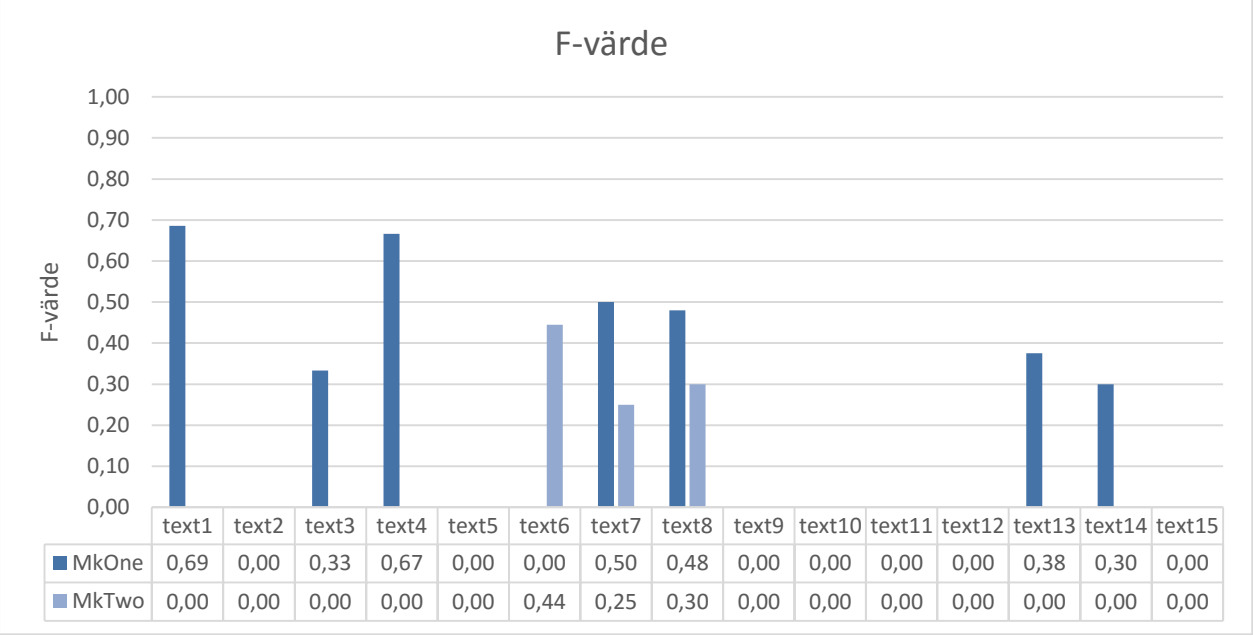
Den andra figuren visar hur hög precision de två metoderna har (se Figur 4). I detta test fick MkOne en genomsnittlig precision på 20 %, det vill säga att en femtedel av dom extraherade meningarna tillhör dom fördefinierade. MkTwo hade en precision på 5 % (se Figur 6).



	text1	text2	text3	text4	text5	text6	text7	text8	text9	text10	text11	text12	text13	text14	text15
MkOne	0,60	0,00	0,33	0,67	0,00	0,00	0,50	0,40	0,00	0,00	0,00	0,00	0,25	0,20	0,00
MkTwo	0,00	0,00	0,00	0,00	0,00	0,33	0,25	0,20	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Figur 4: Mätningar för de två metodernas precision på test-dokumenten

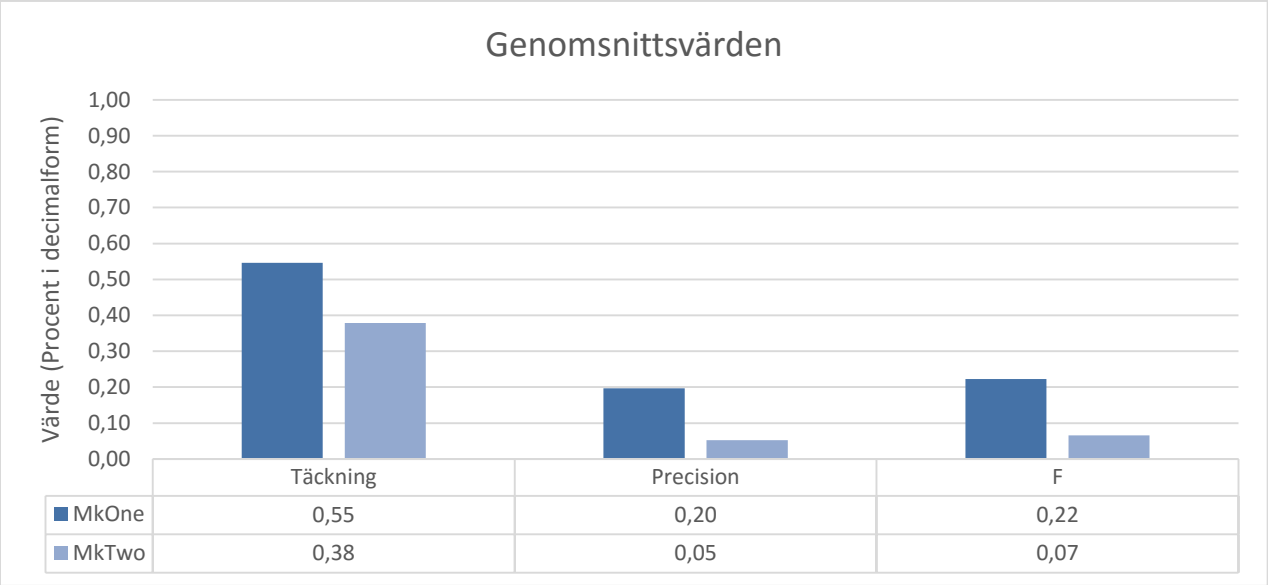
Den tredje figuren visar F-värdet för metoderna, ett snittvärde mellan täckning och precision som indikerar hur bra metoderna prestera. MkOne har ett genomsnittligt F-värde på 0.22 och MkTwo på 0.07 (se Figur 6).



	text1	text2	text3	text4	text5	text6	text7	text8	text9	text10	text11	text12	text13	text14	text15
MkOne	0,69	0,00	0,33	0,67	0,00	0,00	0,50	0,48	0,00	0,00	0,00	0,00	0,38	0,30	0,00
MkTwo	0,00	0,00	0,00	0,00	0,00	0,44	0,25	0,30	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Figur 5: Mätningar för de två metodernas F-värde på test-dokumenten

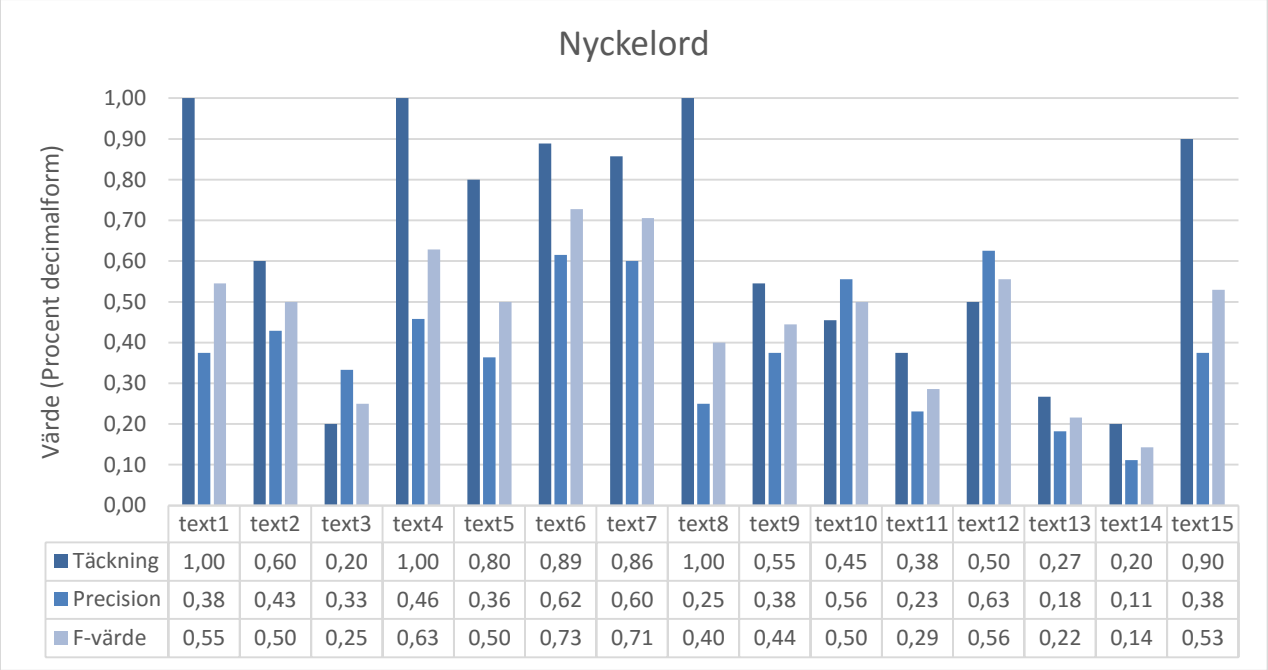
Figuren på nästa sida (se Figur 6) anger genomsnittsvärdena för de två sammanfattnings-metodernas täckning, precision och F-värde.



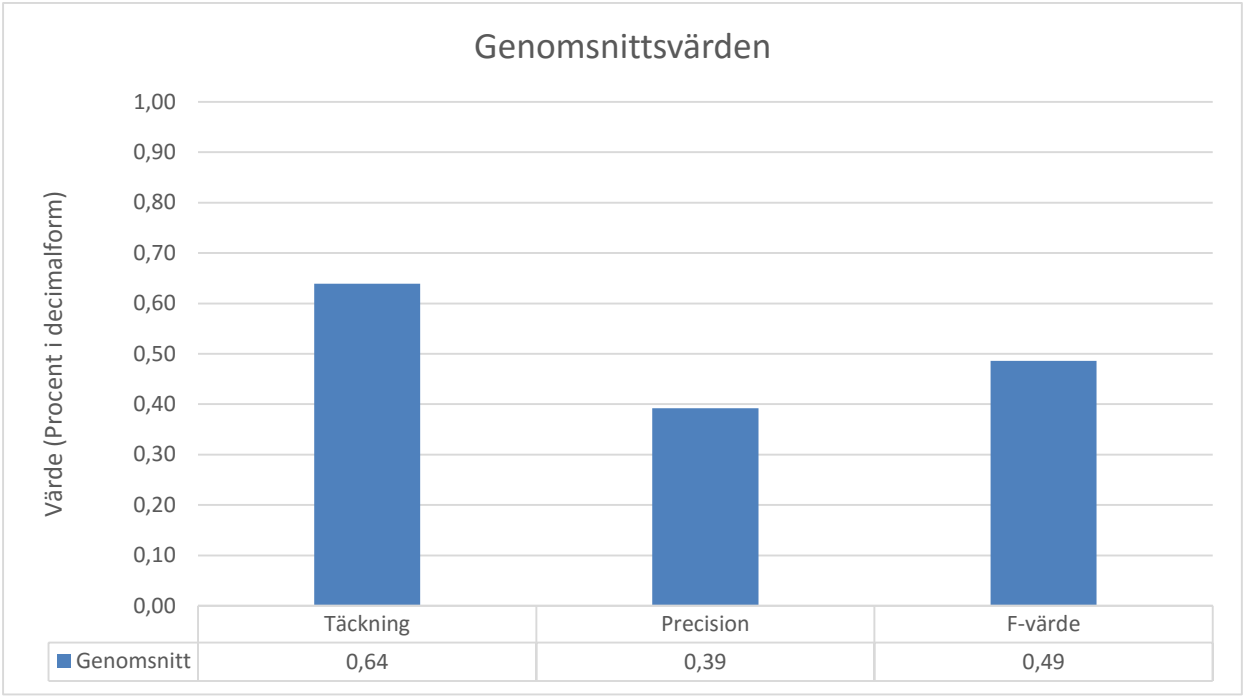
Figur 6: Genomsnittsvärdens för de två metodernas resultat

3.10.2 Sammanställning för hittande av nyckelord

Figuren nedan (se Figur 7) anger metodens täckning, precision och F-värde för varje test-dokument. Metoden hade en genomsnittstäckning på 64 %, genomsnittsprecision på 39 % och ett genomsnittligt F-värde på 0.49 (se Figur 8).



Figur 7: Mätningar för extrahering av nyckelord från test-dokumenten



Figur 8: Genomsnittsvärden för extrahering av nyckelord

4 Analys och diskussion

I detta kapitel kommer arbetet, prototyperna och resultatet att analyseras och diskuteras. För projektet skapades två prototyper med syfte att sammanfatta en text och extrahera nyckelord samt identifiera språk. För att analysera resultatet och få en bild av hur prototyperna klarade deras uppgift skapades flera visuella grafer (se 3.10 Sammanställning av resultat). En kort förklaring till följande gjordes (se 4.1.1 Sammanfattningsmetoderna), med syfte att ge inblick i varför resultatet blev som det blev. För prototyperna MkOne och MkTwo gjordes en graf över deras förmåga att hämta relevant data, hur väl dom täcker de fördefinierade meningarna. För precisionen gjordes en graf över prototypernas förmåga att rangordna data i relevant ordning, sista grafen visar genomsnittsvärdet mellan dessa två tester i form av ett F-värde.

För skapande av nyckelordlistan kombineras två metoder, en metod för frekventa termer och en NER-metod hämtad från OpenNLP. För att testa dom gjordes en fördefinierad lista av ord för varje text, listan representerade relevant data för respektive texter. Metodernas förmåga att hitta dessa ord redovisades i en graf över deras täckning, precision och F-värde (se 4.1.2 Nyckelordlistan).

För att få en bild av hur projektet utvecklades och vad en framtidig strategi kan ge för resultat för Widespace gjordes en förklaring av detta under diskussionen (se 4.2 Diskussion).

4.1 Resultatanalys

4.1.1 Sammanfattningsmetoderna

Det första testet mätte prototypernas täckning för sammanfattningen (se Figur 3). Det vill säga deras förmåga att hitta så många av dom fördefinierade meningarna som möjligt. Enligt grafen finns det en variation i resultatet för både MkOne och MkTwo.

För MkOne erhålls den bästa täckningen på 80 % för text1 (se Bilaga 1), fyra utav de fem fördefinierade meningarna hittades. Lägsta värdet var på 25 % det vill säga en utav fyra meningar hittades för text11 (se Bilaga 2). Ett bra resultat för MkOne byggs på hur mycket frekventa termer som används i relevanta meningar och hur högt meningen är placerat i texten. Det vill säga att varje mening viktas utifrån hur många relevanta ord som meningen innehåller, samt meningens position. Efter viktningen kommer alla meningar att rangordnas, hög vikt betyder hög relevans. Meningarna som valdes ut av MkOne för text1 var mening ett, tre, sex, fem och åtta i relevant ordning (se Tabell 2). Anledningen till att mening ett valdes av MkOne var till största del dess position. Det fanns 15 meningar i texten och första meningen kommer då att viktas med värdet 15. Mening ett innehöll även ett frekvent ord och kommer då att viktas med plus en, meningen erhåller då en vikt på 16. Detta blev den största vikten och därför läggs meningen först i listan. Mening två, fyra och sju blev inte en del av sammanfattningen, detta beror på att mening tre, sex, fem och åtta hade högre vikt.

Tabell 2: Fördefinierade meningar och respektive utvalda meningar av prototyperna MkOne och MkTwo

Fördefinierade	1	3	6	8	13
MkOne	1	3	6	5	8
MkTwo	6	9	3	1	8

För MkTwo erhålls det bästa värdet också för text1 med en täckning på 80 %. Sämsta värdet hämtades från text11 och text12 med en täckning på 0 %. Ingen av de meningar som hämtades tillhörde de fördefinierade meningarna. För MkTwo syns det tydligt i tabellen (se Tabell 2) att det inte finns samma ökning för meningarnas ordningsnummer som för MkOne. Detta beror på att MkTwo inte ser upp för meningarnas position i texten. Varje mening viktas individuellt utifrån deras innehåll av substantiv och subjekt.

Sammanfattningsmetoderna hittar samma antal meningar som det finns fördefinierat vilket begränsar deras tolerans för fel. Om en irrelevant mening hittas, kommer den att inta en plats i listan. Eftersom täckningen ignorerar irrelevanta meningar och bara fokuserar på hur många relevanta meningar som hämtas begränsas sammanfattningsmetodernas förmåga att hålla en bra täckning. Anledningen till denna begränsning är för att inte tappa sammanfattningens syfte, att vara kort och konkret.

Figur 4 visar precisionsmätningen för sammanfattningsmetoderna i MkOne och MkTwo. Det syns att båda metoderna erhåller ett dåligt resultat för precision. Detta beror till största del på hur mätningen för precisionen hanterades. För sammanfattningen var det inte möjligt att mäta precisionen utifrån förmågan att bara hitta relevant data. Detta beror på att metoderna extraherar exakt det antal meningar som fanns fördefinierade. Precisionen valdes istället att byggas på hur väl meningarna hittas i rätt ordning. Det vill säga för att metoderna ska vara precisa måste de fördefinierade meningarna extraheras samt rangordnas i relevant ordning. Genomsnittet för precisionen av MkOne var på 20 % vilket är fyra gånger bättre än MkTwo. Detta beror på att MkOne tog hänsyn till meningarnas position i texten, hög placering desto högre relevans. Det visade sig även att tpersonernas val av meningar i texten låg bland de översta i texterna och resulterade i att MkOne hade bättre precision.

För att få en bild av metodernas förmåga att skapa en bra sammanfattning användes F-måttet (se Figur 5), ett genomsnittsvärde mellan täckningen och precisionen. Det syns väldigt tydligt att båda metoderna inte gav ett bra resultat, dock erhöll metoderna några fall som resulterade bra. Detta beror på att precisionen för dessa var större än 0 %. Formeln för F-måttet gör att täckningen och precisionen är beroende av varandra, en multiplicering i täljaren (se ekvation 6) och tolererar inte ett nollvärde.

Syftet med dessa tester är att generalisera metodernas prestation, hitta ett snittvärde mellan dess förmåga att hitta relevant data och i detta fall rangordna hämtad data rätt. Problemet med detta förknippades med dess förmåga att rangordna rätt, en svår process för både MkOne och MkTwo. För projektet ansågs täckningsvärdet ha störst betydelse och kommer användas som det redovisande resultatet för prototyperna.

4.1.2 Nyckelordlistan

I Figur 7 visas precision, täckning och F-värdet för nyckelordlistan. För att skapa listan användes två metoder, en termfrekvens-metod och en NER-metod. Syftet med denna kombination är att hitta relevanta data i form av nyckelord. Det visade sig att båda metoder hade sina fördelar och nackdelar. Termfrekvensen hittade ord som ansågs vara relevanta för textens tema, upprepade termer. Problemet med detta är att ord som inte har relevans för texten även kan upprepas och läggas i ordlistan, vilket påverkade precisionen för metoden. NER-metoden hittade alla namn och organisations namn,

eftersom att alla organisationer hittades kommer även dom som inte har en relevans för texten ingå. Detta betyder att precisionen kommer att sänkas.

För text1 (se Bilaga 1) hittades alla nyckelorden och ett täckningsvärde på 100 % erhölls. För att uppnå detta krävdes en kombination av båda metoderna. Fem utav dom nio fördefinierade orden hittades av termfrekvens-metoden (se Tabell 3), de resterande fyra orden hittades av NER metoden (se Tabell 3). Fördelen med en bred täckning hade också en nackdel, precisionen resulterade i 38 % (se Figur 7) och sänkte nyckelordlistans F-värde. Detta beror på att båda metoderna även hittade irrelevanta ord. För termfrekvensen hittades sju övriga ord och för NER-metoden hämtades åtta (se

Tabell 4). För att minska detta kan termfrekvensen ökas, det vill säga ord måste upprepas mer än två gånger för att hamna i listan. Detta kan även leda till att relevanta ord missas och valdes inte att användas.

Tabell 3: Lista över fördefinierade ord, hämtade ord av Termfrekvens och NER för text1

Fördefinierade ord	**Termfrekvens**	**NER**
Agency	agency	
Dwight D. Eisenhower		Dwight D. Eisenhower
NACA		NACA
Nasa	nasa	NASA
National Advisory Committee for Aeronautics		National Advisory Committee for Aeronautics
National Aeronautics and Space Administration		National Aeronautics and Space Administration
Research	research	
Science	science	
Space	space	

Tabell 4: Irrelevanta ord hämtade av termfrekvens och NER för text1

Termfrekvens	**NER**
1958	Heliophysics Research Program
aeronaut	International Space Station
launch	Launch Services Program

mission	Space Launch System
nation	Science Mission Directorate
program	Space Shuttle
System	

4.2 Diskussion

Projektet initierades med syfte att underlätta sortering av inkommande ärenden genom att utifrån textinnehåll klassificera den till en specifik avdelning. Om en text innehöll tekniska fraser, skulle den klassas som en teknisk fråga och tilldelas till tekniska avdelningen. Om texten innehöll information om kampanjer och företag skulle detta även synas som metadata om texten. Anledningen till detta var att Widespace bara hade tillgång till en mailbox, alla ärenden kom till samma mailbox och skulle sedan sorteras ut till respektive avdelning. Målet var då att träna en modell till att klassificera text och skapa metadata med syfte att underlätta sorteringen till respektive avdelningar hos Widespace. Men som projektet gick igång kom de nya krav från Widespace, ett förslag blev presenterat. Istället för att sorteringen sker hos Widespace, skulle det göras möjligt att skapa flera mailboxar. En mailbox för varje avdelning, sorteringen sker då redan från kundens sida och Widespace kan spara resurser. Detta gav projektet en liten vändning, det var inte längre nödvändigt att klassificera. Oberoende av avdelning krävdes det fortfarande att varje ärende skulle kunna tolkas och tydas bra. Mängden av ärenden som hanteras varje vecka ligger runt hundratals och är fortfarande en tidskrävande uppgift. För att hjälpa Widespace med denna hantering skapades nya mål för projektet. Syftet med dessa mål var att skapa verktyg för att underlätta varje anställds förmåga att tolka ett nytt eller gammalt ärende bättre. En sammanfattande information om vad ärendet omfattar i form av metadata. Detta skulle göras som en automatiserad lösning som även kunna kompletteras eller ändras med manuell inmatning. De tre verktyg som togs fram var en automatiserad sammanfattning av ärendets innehåll, en nyckelordlista och en indikation om vilket språk som ärendet är skrivet i.

Som vi gick igång med utvecklingen av verktygen träffade vi på nya problem. Det visade sig att det inte var möjligt att extrahera tidigare ärenden från deras nuvarande system. Syftet med denna extrahering var att använda tidigare ärenden som testdata, data som verktygen kommer att användas med i framtidiga bruk. För att kunna fortsätta projektet valdes då att använda Wikipedia-texter som testdata. Med färdiga testverktyg kunde ett resultat redovisas, det framstod att prototypen MkOne visade det bästa resultatet för sammanfattningsmetoden. Det förekom bara ett problem med detta resultat, testningen av metoden. Testdatat som resultatet bygger på är via Wikipedia-texter och inte data som prototyperna kommer användas med. Det som skiljer Wikipedia-texter och ärenden ligger i hur ett ärende är skrivet samt dess variation i längd. Ett ärende kan förekomma i stor variation, detta beror helt på personen som skapar ärendet. Det kan vara svårt att sammanfatta ett ärende som innehåller fem meningar, som alla handlar om olika saker. Det kan däremot vara lättare att sammanfatta ett ärende som omfattar en sak. Detta problem definierar syftet med nyckelordlistan. Med en kombination av både en sammanfattning och en nyckelordlista kan ärenden av alla variationer hanteras bättre. En tolerans för hur kort en text får vara för att skapa en sammanfattning kan ställas. Nyckelordslistan kommer då att fungera som en kompletterande sammanfattning för korta texter, men även för längre texter tillsammans med sammanfattningen. För texter som omfattar många olika saker kommer nyckelordlistan även att fungera som en summering. Eftersom det inte behöver finnas en länkning

mellan varje ord i listan, kommer den att fungera oberoende av hur många olika teman som texten innehåller.

4.3 Externa aspekter

Metoden har utvecklats med framtidig utveckling i åtanke. Detta innebar att metoden modellerades modulärt vilket gör att framtida förändringar av metoden är simpla att genomföra och det är även möjligt att på ett enkelt sätt byta ut hela delar av modellen.

Resultatet som erhölls från testerna av metoden påvisar att en viss ekonomiskavlastning kan uppnås genom att implementera den utvecklade metoden i ett ärendehanteringssystem. Detta då metoden kan spara både tid och resurser genom att låta de anställda fokusera på ärendena och spendera mindre tid på att genomföra administrativt arbete, arbete som metoden istället kan sköta.

Arbetet med detta projekt har hållit sig inom de ramar som finns vad gäller miljö och etik. Även den utvecklade metoden håller sig inom dessa ramar.

5 Slutsatser

En fungerande modell togs fram som har visat möjligheterna att automatisk utvinna metadata från texter. Målet för de framtagna prototyperna samt undersökning av de metoder och algoritmer som låg till grund för implementationen uppnåddes. Den slutgiltiga egenutvecklade prototypen MkOne innehöll metoder som genererade en sammanfattning, en nyckelordlista och en indikation om vilket språk texten är skriven i. Resultatet visade en genomsnittlig täckning på 55 % för sammanfattningen, lite över hälften av alla meningar hittades och en precision på 20 % erhölls. För nyckelordlistan hämtades ett genomsnitts resultat för täckningen på 64 % och 39 % för precisionen. Med fortsatt arbete kring träning av modeller och lingvistiska metoder kan den utvecklade prototypen förbättras till en grad att en implementation kan göras och användas för ett ärendesystem åt Widespace.

5.1 Framtida syften och förbättringar

Framtida syften med prototypen är att skapa en bättre sökförmåga för Widespace. Med flera hundratals ärenden varje vecka kommer det ställa till problem att leta upp ärenden. I dagslägets system kan det bara göras sökningar i form av ja och nej, det vill säga finns ärendet mellan dessa datum, har Dennis skapat det, har ärendet status “Färdig” osv. Detta är en stor fördel om man har specifik data om ärendet, men om en anställd bara kommer ihåg delar av detta kan det resultera i väldigt många ärenden. Det ska vara möjligt för anställda att kunna söka på ord, likt en sökning via Google, en vektorbaserad sökning. För att lösa detta kommer nyckelordlistan till användning. Ärenden rankas utifrån deras relevans till sökningen. Detta kombinerat med den redan existerande sökfunktionen kommer anställda lättare åt arkiverade ärenden.

Källförteckning

[1] Gobinda G. Chowdhury, “Annual Review of information science and technology: Natural Language Processing”, 2003; s. 51-89.

[2] James F. Allen, “Encyclopedia of Computer Science: Natutal Language Processing”, ACM Digital Library, 2003; s. 1218-1222.

[3] Indurkhya N, Damerau JF, “Handbook of Natural Language Processing, Second Edition”, 2010.

[4] Collobert, Weston, Bottou, Karlen, Kavukcuoglu, kuksa, “Natural Language Processing (Almost) from scratch”, *Journal of Machine Learning Research*, vol. 12, 2011, s. 2493-2537

[5] Eric F. Tjong Kim Sang, Sabine Buchholz, “Introduction to the CoNLL-2000 shared task: chunking”, ConLL '00 Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning, vol. 7, 2000, s. 127-132

[6] Xavier Carreras, Llúıs Marquez, “Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling”, *CONLL '05 Proceedings of the Ninth Conference on Computational Natural Language Learning*, 2005, s. 152-164

[7] MuntsaPadro, Lluıs Padro,”Comparing methods for language identification”

[8] Ronen Feldman,James Sanger, ”The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data”, 2006

[9] Cambridge University Press, “Information Retrieval”, 2009

[10] Ian H. Witten, Eibe Fran, “Data Mining: Practical Machine Learning Tools and Techniques, Second Edition”, 2005

[11] NISO (National Information Standards Organization), “Understanding Metadata”, http://www.niso.org/publications/press/UnderstandingMetadata.pdf, Publicerad 2004. Hämtad 2015-09-30.

[12] NISO (National Information Standards Organization), “Metadata Demystified”, http://www.niso.org/standards/resources/Metadata_Demystified.pdf, Publicerad 2003-07. Hämtad 2015-09-30.

[13] DCMI (Dublin Core Metadata Initiative), ”User Guide”, http://dublincore.org/documents/usageguide/, Ändrad 2011-09-06. Hämtad 2015-10-01.

[14] S. Weibel, J. Kunze, C. Lagoze, M. Wolf, “Dublin Core Metadata for Resource Discovery”, http://www.rfc-editor.org/info/rfc2413, Publicerad 1998-09. Hämtad 2015-09-30.

[15] Nancy Ide, “Encoding standards for large text resources: the Text Encoding Initiative”, *COLING '94 Proceedings of the 15th conference on Computational linguistics*, vol. 1, 1994, s. 574-578.

[16] C. M. Sperberg-McQueen, Lou Burnard, “The Design of the TEI Encoding Scheme”, *Computers and the Humanities*, vol. 29, nr 1, 1995, s. 17-39

[17] Linda Cantara, “METS: The Metadata Encoding and Transmission Standard”, *Cataloging & Classification Quarterly*, vol. 40, nr. 3/4, 2005, s. 237-253

[18] Richard Gartner, “METS: Metadata Encoding and Transmission Standard” , http://greenstonesupport.iimk.ac.in/greenstone2010-nepal/sample_files/Word_and_PDF/difficult_pdf/pdf06-weirdchars.pdf, Publicerad 2002-10. Hämtad 2015-10-05

[19] Rebecca Guenther, Sally McCallum, “New Metadata Standards for Digital Resources: MODS and METS”, *Bulletin of the American Society for Information Science and Technology*, vol. 29, nr. 2, 2003, s. 12–15

[20] Sally H. McCallum, “An introduction to the Metadata Object Description Schema (MODS)”, *Library Hi Tech*, vol. 22, nr 1, 2004, s. 82 - 88

[21] L. R. .Rabiner, B. H. Juang, “An introduction to hidden Markov models”, *ASSP Magazine, IEEE*, vol. 3, nr. 1, 1986, s. 4-16

[22] Sean R Eddy,”Hidden Markov models”, *Current Opinion in Structural Biology*, vol. 6, nr. 3, 1996, s. 361–365

[23] Phil Blunso, “Hidden Markov Models”, http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf, Publicerad 2004-08-14. Hämtad 2015-10-07.

[24] Stephen Robertson, ”Understanding inverse document frequency: on theoretical arguments for IDF”, *Journal of Documentation*, vol. 60, nr. 5, 2004, s. 503 – 520

[25] Brian Lott, “Survey of Keyword Extraction Techniques”, http://www.cs.unm.edu/~pdevineni/papers/Lott.pdf Publicerad 2012-12-04. Hämtad 2015-10-07.

[26] I. Dan Melamed, Ryan Green, Joseph P. Turian, ”Precision and Recall of Machine Translation”, *NAACL-Short '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003 -- short papers*, vol. 2, 2003 , s. 61-63

[27] Yutaka Sasaki, “The truth of the F-measure”, 2007

[28] Ayodele T., Khusainov R., Ndzi D.,”Email Classification and Summarization: A Machine Learning Approach”, *Wireless, Mobile and Sensor Networks, 2007. (CCWMSN07). IET Conference on*, 2007, s. 805 - 808

[29] Shamsa R., Hashem M.M.A., Hossain A., Akter S.R., Gope, M., ” Corpus-based web document summarization using statistical and linguistic approach”, *Computer and Communication Engineering (ICCCE), 2010 International Conference on*, 2010, s. 1-6

[30] Pal A.R., Saha D., "An approach to automatic text summarization using WordNet", *Advance Computing Conference (IACC), 2014 IEEE International*, 2014, s. 1169 - 1173

[31] Ranks NL, “Stopwords”, http://www.ranks.nl/stopwords, Hämtad 2015-11-02

Bilagor

Bilaga 1: Text1 (uppdelad i numrerade meningar)

1. The National Aeronautics and Space Administration (NASA) is the United States government agency responsible for the civilian space program as well as aeronautics and aerospace research.
2. President Dwight D. Eisenhower established the National Aeronautics and Space Administration (NASA) in 1958 with a distinctly civilian (rather than military) orientation encouraging peaceful applications in space science.
3. The National Aeronautics and Space Act was passed on July 29, 1958, disestablishing NASA's predecessor, the National Advisory Committee for Aeronautics (NACA).
4. The new agency became operational on October 1, 1958.
5. Since that time, most US space exploration efforts have been led by NASA, including the Apollo moon-landing missions, the Skylab space station, and later the Space Shuttle.
6. Currently, NASA is supporting the International Space Station and is overseeing the development of the Orion Multi-Purpose Crew Vehicle, the Space Launch System and Commercial Crew vehicles.
7. The agency is also responsible for the Launch Services Program (LSP) which provides oversight of launch operations and countdown management for unmanned NASA launches.
8. NASA science is focused on better understanding Earth through the Earth Observing System, advancing heliophysics through the efforts of the Science Mission Directorate's Heliophysics Research Program, exploring bodies throughout the Solar System with advanced robotic spacecraft missions such as New Horizons, and researching astrophysics topics, such as the Big Bang, through the Great Observatories and associated programs.
9. NASA shares data with various national and international organizations such as from the Greenhouse Gases Observing Satellite.

Bilaga 2: Text11 (uppdelad i numrerade meningar)

1. Coca-Cola is a carbonated soft drink.
2. It is produced by The Coca-Cola Company of Atlanta, Georgia, and is often referred to simply as Coke (a registered trademark of The Coca-Cola Company in the United States since March 27, 1944).
3. Originally intended as a patent medicine when it was invented in the late 19th century by John Pemberton, Coca-Cola was bought out by businessman Asa Griggs Candler, whose marketing tactics led Coke to its dominance of the world soft-drink market throughout the 20th century.
4. The name refers to two of its original ingredients: kola nuts, a source of caffeine, and coca leaves.
5. The current formula of Coca-Cola remains a trade secret, although a variety of reported recipes and experimental recreations have been published.
6. The company produces concentrate, which is then sold to licensed Coca-Cola bottlers throughout the world.
7. The bottlers, who hold territorially exclusive contracts with the company, produce finished product in cans and bottles from the concentrate in combination with filtered water and sweeteners.
8. The bottlers then sell, distribute and merchandise Coca-Cola to retail stores, restaurants and vending machines.
9. The Coca-Cola Company also sells concentrate for soda fountains to major restaurants and food service distributors.
10. The Coca-Cola Company has, on occasion, introduced other cola drinks under the Coke brand name.
11. The most common of these is Diet Coke, with others including Caffeine-Free Coca-Cola, Diet Coke Caffeine-Free, Coca-Cola Cherry, Coca-Cola Zero, Coca-Cola Vanilla, and special versions with lemon, lime, or coffee.

12. In 2013, Coke products could be found in over 200 countries worldwide, with consumers downing more than 1.8 billion company beverage servings each day.
13. Based on Interbrand's best global brand study of 2015, Coca-Cola was the world's third most valuable brand.