



Högskoleingenjörsutbildning i datateknik

Grundnivå, 15hp

# **AI-stöd för kvalitetssäkring och informationssökning i teknisk dokumentation**

**En jämförelse mellan regelbaserade och AI-baserade metoder för GDPR-detektion och semantisk sökning**

**TUSHAN BARUA  
MARTIN DAOUD**



# **AI-stöd för kvalitetssäkring och informationssökning i teknisk dokumentation**

En jämförelse mellan regelbaserade och AI-baserade metoder för GDPR-detektion och semantisk sökning

## **AI-assisted quality assurance and information retrieval in technical documentation**

A comparison between rule-based and AI-based methods for GDPR detection and semantic search

TUSHAN BARUA  
MARTIN DAOUD

Examensarbete inom datateknik  
Grundnivå, 15 hp

Handledare på KTH: Bertil Guve  
Examinator: Anders Lindström  
TRITA-CBH-GRU-2026:131

KTH  
Skolan för kemi, bioteknologi och hälsa  
141 52 Huddinge, Sverige



## Sammanfattning

I IT-organisationer växer mängden intern dokumentation kontinuerligt, vilket medför två utbredda problem: känslig information exponeras oavsiktligt i löpande text och befintliga söksystem hittar inte relevant innehåll om exakt rätt ord inte används. Traditionella verktyg klarar inte av att hantera dessa problem eftersom de saknar förmåga att tolka sammanhang. Detta examensarbete undersöker hur AI-baserade metoder kan användas för att automatisera identifiering av känslig information och förbättra informationsåtervinning i teknisk dokumentation lagrad i wiki-miljöer. Arbetet är utfört hos IT-tjänsteföretaget Axians.

Arbetet utvärderar tre metoder för identifiering av känslig information: reguljära uttryck, Microsoft Presidio och en generativ språkmodell, samt jämför nyckelordsbaserad sökning med semantisk sökning och en hybridmetod. Metoderna testades på syntetiskt genererade dokument som efterliknar teknisk IT-dokumentation på svenska och engelska. Resultaten från det syntetiska testsetet indikerar att den generativa språkmodellen Mistral 7B identifierar känslig information med hög träffsäkerhet, medan de regelbaserade metoderna missar en stor del av de kontextberoende entiteterna. För sökning uppnådde det semantiska systemet bättre resultat i det använda testsetet, särskilt på synonymfrågor och tvärspråkliga sökningar. Slutsatsen är att AI-baserade metoder har stor potential att förbättra både datasäkerhet och informationstillgänglighet i tekniska dokumentationsmiljöer.

### Nyckelord

GDPR-detektion, semantisk sökning, teknisk dokumentation, stora språkmodeller, informationsåtervinning, AI, wiki, regelbaserade metoder, RAG, Microsoft Presidio



## **Abstract**

In IT organizations, the volume of internal documentation grows continuously, giving rise to two widespread problems: sensitive information is unintentionally exposed in plain text, and existing search functionality fails to retrieve relevant content unless exact terminology is used. Traditional tools are insufficient for addressing these problems as they lack the ability to interpret context. This thesis investigates how AI-based methods can be used to automate the identification of sensitive information and improve information retrieval in technical documentation stored in wiki environments. The work was conducted at the IT services company Axians.

The study evaluates three methods for identifying sensitive information: regular expressions, Microsoft Presidio and a generative language model, and compares keyword-based search with semantic search and a hybrid method. The methods were tested on synthetically generated documents resembling technical IT documentation in Swedish and English. The results from the synthetic test set indicate that the generative language model Mistral 7B identifies sensitive information with high recall, while the rule-based methods miss a substantial portion of the context-dependent entities. For information retrieval, the semantic system achieved better results in the evaluated test set, particularly on synonym queries and cross-lingual searches. The conclusion is that AI-based methods have significant potential to improve both data security and information accessibility in technical documentation environments.

## **Keywords**

GDPR detection, semantic search, technical documentation, large language models, information retrieval, AI, wiki, rule-based methods, embeddings, Microsoft Presidio



## Förord

Denna rapport markerar avslutet av högskoleingenjörsprogrammet i datateknik på KTH, Kungliga Tekniska högskolan.

Vi vill rikta ett stort tack till Bertil Guve, handledare på KTH, för värdefull vägledning och feedback genom hela arbetet. Hans engagemang och tekniska insikter har varit till stor hjälp både vid utformningen av undersökningen och vid skrivandet av denna rapport.

Vi vill också tacka Christopher Serre på Axians för att du tog dig tid, delade med dig av din kunskap och gav oss möjligheten att genomföra arbetet i en verklig miljö. Utan tillgången till Axians FosWiki-miljö och din erfarenhet av de praktiska utmaningarna hade detta arbete inte varit möjligt.

Stockholm, 2026

Tushan Barua & Martin Daoud



# Innehållsförteckning

<b>1</b>	<b>Inledning</b> .....	<b>1</b>
1.1	Problemformulering.....	1
1.2	Målsättning.....	2
1.3	Avgränsningar .....	2
1.4	Författarnas bidrag till examensarbetet .....	3
1.5	Etiskt förhållningssätt .....	3
<b>2</b>	<b>Teori och bakgrund</b> .....	<b>5</b>
2.1	Teknisk dokumentation och dess utmaningar.....	5
2.2	GDPR och identifiering av känslig information .....	5
2.3	Semantisk sökning och RAG .....	7
2.4	Lokala språkmodeller och on-prem AI .....	8
2.5	Tidigare Arbeten.....	8
2.5.1	Anonymisation Models for Text Data (Lison et al., 2021) .....	8
2.5.2	Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis et al., 2021) .....	9
2.5.3	Analyzing the Efficacy of Microsoft Presidio in Identifying Social Security Numbers in Unstructured Text (Friebely) .....	9
<b>3</b>	<b>Metod</b> .....	<b>11</b>
3.1	Metoder för GDPR-detektion.....	11
3.1.1	Översikt.....	11
3.1.2	Testset .....	11
3.1.3	Visuellt exempel på testdata.....	13
3.1.4	Metod 1 - Regex .....	13
3.1.5	Metod 2 - Microsoft Presidio.....	13
3.1.6	Metod 3 - LLM via Mistral API.....	14
3.1.7	Utvärderingsmetodik.....	14
3.1.8	Implementationsdetaljer .....	15
3.2	Sökmeter.....	15
3.2.1	Översikt.....	15
3.2.2	Testset .....	16
3.2.3	System A - Syntaktisk sökning med BM25 .....	17

3.2.4	System B - Semantisk sökning med sentence-transformers och FAISS	17
3.2.5	System C - Hybridmetod med Reciprocal Rank Fusion .....	18
3.2.6	Utvärderingsmetodik .....	18
3.2.7	Implementationsdetaljer .....	19
3.3	Testmiljö och hårdvara .....	19
<b>4</b>	<b>Resultat .....</b>	<b>23</b>
4.1	GDPR-detektion .....	23
4.1.1	Aggregerade mätvärden .....	23
4.1.2	Prestanda per entitetstyp .....	24
4.1.3	Resultat per dokument .....	24
4.1.4	Falska positiva och falska negativa .....	24
4.2	Semantisk sökning .....	24
4.2.1	Aggregerade mätvärden .....	24
4.2.2	Resultat per frågetyp .....	25
<b>5</b>	<b>Analys och diskussion.....</b>	<b>27</b>
5.1	Tolkning av GDPR-detektionsresultaten.....	27
5.2	Tolkning av sökresultaten .....	27
5.3	Alternativa lösningar.....	28
5.4	Ekonomiska, sociala, etiska och miljömässiga aspekter .....	28
<b>6</b>	<b>Slutsatser .....</b>	<b>31</b>
6.1	Arbetets bidrag och resultat.....	31
6.2	Reflektion och rekommendationer .....	32
6.3	Fortsatt arbete.....	32
	<b>Källförteckning .....</b>	<b>33</b>
	<b>Bilagor .....</b>	<b>37</b>
	Bilaga A – Mätvärden för GDPR-detektion .....	37
	Bilaga B – Mätvärden för söksystem .....	43



# 1 Inledning

Axians är ett svenskt IT-tjänsteföretag som arbetar med drift och förvaltning av komplexa IT- och telekomlösningar. Företaget vänder sig till kunder inom både offentlig och privat sektor och erbjuder tjänster inom bland annat cybersäkerhet, molnlösningar och nätverksinfrastruktur. Inom organisationen används en wiki-baserad plattform för att lagra teknisk dokumentation såsom systemöversikter, driftsrutiner och konfigurationsbeskrivningar.

Mängden intern teknisk dokumentation i IT-organisationer har ökat kontinuerligt i takt med digitaliseringen, vilket har lett till ökade krav på både informationssökning och informationssäkerhet. Dokumentationen är en central del av IT-drift och används både i det dagliga arbetet och vid onboarding av ny personal.

I denna typ av miljö uppstår två återkommande problem. För det första kan känslig information förekomma i dokumentationen utan att det upptäcks manuellt. För det andra har traditionella sökfunktioner begränsad förmåga att hitta relevant information när andra begrepp eller språk används för att beskriva samma sak.

## 1.1 Problemformulering

Teknisk dokumentation i IT-organisationer växer kontinuerligt och blir allt svårare att hantera effektivt. Enligt handledaren Christopher Serre uppstår två centrala problem i Axians Foswiki: oavsiktlig förekomst av känslig information i dokumentationen samt begränsad sökbarhet i befintliga system. Dokumentationen innehåller information om nätverkskonfigurationer, systemuppsättningar och driftsrutiner som är nödvändiga för drift och support. I praktiken är innehållet ofta ojämnt strukturerat och uppdateras sällan, vilket försvårar både underhåll och informationssökning enligt handledaren.

Känslig information kan ibland förekomma i tekniska dokument, exempelvis i form av lösenord, API-nycklar eller tjänstekonton i klartext. Sådan information är svår att identifiera manuellt i stora textmängder och kan utgöra både en säkerhetsrisk och ett potentiellt GDPR-problem.

Samtidigt bygger sökfunktionen i FosWiki på syntaktisk (lexikal) matchning, vilket innebär att endast exakta ordträffar returneras. Detta leder till att relevant information kan missas om andra termer eller språk används, exempelvis mellan "brandvägg" och "firewall" eller relaterade begrepp som "ACL-regler".

Molnbaserade AI-lösningar kan potentiellt förbättra både informationssökning och detektion av känslig information. Dessa är dock inte alltid lämpliga i miljöer där dokumentationen innehåller intern infrastrukturinformation, eftersom överföring av sådan data till externa tjänster kan strida mot GDPR och interna säkerhetspolicys.

## 1.2 Målsättning

Syftet med examensarbetet är att undersöka hur olika tekniska angreppssätt presterar vid hantering av två centrala problem inom teknisk dokumentation: identifiering av GDPR relevant information samt informationsåtervinning. Studien jämför regelbaserade metoder med metoder baserade på generativ AI samt analyserar skillnader mellan lexikal, semantisk och hybridsökning.

Målet är att genomföra en kvantitativ utvärdering av dessa metodkategorier genom kontrollerade experiment i en testmiljö.

Rapporten undersöker därför två frågor:

**F1:** Hur väl identifierar en språkmodell GDPR-känslig information i teknisk text jämfört med regelbaserade metoder mätt med Precision, Recall och F1-score?

**F2:** Hur mycket förbättras sökkvaliteten när semantisk sökning används istället för syntaktisk sökning mätt med Precision, Recall och MRR?

## 1.3 Avgränsningar

Arbetet avgränsas till teknisk dokumentation lagrad i FosWiki. Testning av regex och semantisk sökning sker i en isolerad labbmiljö utan koppling till externa tjänster eller molntjänster. Mistral 7B utvärderas via ett molnbaserat API, men testerna genomfördes uteslutande mot syntetiskt genererade dokument utan faktisk känslig information. I en produktionsmiljö hos Axians ersätts API-anropet med en lokal Ollama-instans.

Fokus ligger på två problemområden, detektion av känslig information och semantisk sökning, vilka identifierades som mest relevanta i förstudien. Övriga funktioner som taggning och kvalitetsgranskning av dokumentation lämnas utanför den formella utvärderingen. Efter diskussion med handledaren beslutades det om att GDPR-detektion och semantisk sökning var de mest akuta problemen i Axians befintliga FosWiki-miljö och att en fokuserad jämförelse av dessa två ger mer reproducerbara och generaliserbara resultat inom arbetets tidsram.

Retrieval Augmented Generation RAG beskrivs i teoridelen eftersom det utgör det bredare tekniska sammanhang som söksystemet ingår i, men en fullständig RAG pipeline med genererande LLM implementeras inte. Avgränsningen motiveras av att examensarbetet fokuserar på söksteget, att hitta rätt dokument och att det generativa steget skulle kräva ytterligare infrastruktur och utvärderingsmetodik utanför tidsramen.

Testdatat som används i arbetet är primärt på svenska, men inkluderar även engelska dokument för att spegla den faktiska tvåspråkiga karaktären hos Axians FosWiki-miljö. Detektering av känslig information är delvis språkberoende eftersom regelbaserade mönster som regex fungerar oberoende av språk för strukturerade format, medan kontextuell information kräver att modellen förstår det svenska

språket. Resultaten är därför inte nödvändigtvis överförbara till andra språk utan vidare utvärdering.

#### **1.4 Författarnas bidrag till examensarbetet**

Författarna har tillsammans ansvarat för samtliga delar av examensarbetet. Ingen av författarna har haft ett större ansvar än den andra, varken när det gäller det tekniska arbetet eller rapporten.

#### **1.5 Etiskt förhållningssätt**

Rapportens författare intygar härmed att rapporten är framtagen utan hjälp av generativ AI än för andra ändamål än språkkontroll.



## 2 Teori och bakgrund

Detta kapitel behandlar den teoretiska bakgrunden till studien, inklusive centrala begrepp inom teknisk dokumentation, informationssökning och identifiering av känslig information. Vidare presenteras relevanta tekniker och tidigare forskning som ligger till grund för de metoder som utvärderas i arbetet.

### 2.1 Teknisk dokumentation och dess utmaningar

Teknisk dokumentation i IT-organisationer fyller en viktig funktion. Den beskriver hur system är uppsatta, hur nätverk är konfigurerade och vilka rutiner som gäller vid drift och felsökning. När dokumentationen fungerar bra minskar det beroendet av enskilda personer och gör det lättare att sätta in ny personal.

I praktiken är det dock vanligt att dokumentationen inte håller tillräcklig kvalitet, det vill säga att den är ofullständig, föråldrad eller svår att hitta i. Forskning om wikis i organisationer visar att innehållet tenderar att växa organiskt utan styrning, vilket leder till ojämn struktur, dubblerat innehåll och information som inte uppdateras i takt med att systemen förändras [1]. Det innebär att den som söker information inte alltid kan lita på att det den hittar stämmer.

Foswiki är en öppen källkodbaserad wiki-plattform utvecklad för samarbete, kunskapsdelning och strukturerad informationshantering inom organisationer. Plattformen är flexibel och anpassningsbar, och innehållet lagras som textbaserade filer i ett eget markup format Topic Markup Language, TML, vilket möjliggör direkt filåtkomst och programmatisk bearbetning [2][3].

### 2.2 GDPR och identifiering av känslig information

GDPR, den europeiska dataskyddsförordningen, ställer krav på att personuppgifter och känslig information hanteras på ett ansvarsfullt sätt. I teknisk dokumentation är det vanligt att sådan information skrivs in direkt i den löpande texten utan att det uppmärksammas. Det beror på att dokumentation ofta skrivs snabbt och i praktiska sammanhang, till exempel när någon dokumenterar en installation eller en felsökningsprocedur och klistrar in faktiska värden för att göra instruktionen konkret och användbar. Det kan handla om IP-adresser, användarnamn, lösenord, API-nycklar och i vissa fall personuppgifter kopplade till specifika system [4].

Det finns i huvudsak två typer av metoder för att automatiskt identifiera sådan information i text. Den första typen är regelbaserade metoder, där regex är det vanligaste exemplet. Regex matchar text mot fördefinierade mönster, till exempel att ett IP-adressformat ser ut som fyra siffror separerade av punkter. Metoden är snabb och deterministisk, vilket innebär att samma indata alltid ger samma utdata, men kräver manuellt underhåll av mönster och klarar inte av att tolka sammanhang [5]. En mening som "lösenordet är katt123" innehåller inget fast mönster att matcha mot och missas därför helt. Tabell 2.1 visar exempel på regex-mönster för de vanligaste entitetstyperna i teknisk IT-dokumentation, och illustrerar tydligt varför lösenord utan fast format inte kan identifieras.

Tabell 2.1: Regex-mönster för vanliga entitetstyper i teknisk IT-dokumentation. Lösenord (sista raden) saknar fast format och kan inte identifieras med regelbaserade metoder.

Entitetstyp	Regex-mönster	Exempelmatchning
IP-adress	<code>\b\d{1,3}(\.\d{1,3}){3}\b</code>	192.168.1.45
E-postadress	<code>[\w.+-]+@[ \w-]+\.[a-zA-Z]{2,}</code>	Erik@gmail.com
Personnummer	<code>\d{6}-\d{4}</code>	850612-1234
Lösenord	(inget fast mönster - kan ej matchas)	katt123 (matchar ej)

Den andra typen är modellbaserade metoder. Named Entity Recognition (NER) är en teknik där en modell tränas att känna igen och klassificera entiteter i text, till exempel personer, platser och organisationer. NER-baserade verktyg som Microsoft Presidio kombinerar regex med NER och kontextmedveten förstärkning, vilket gör det möjligt att identifiera känslig information som regelbaserade metoder missar, till exempel när känslig information framgår av sammanhanget snarare än av ett fast mönster [6]. Presidio använder spaCy som NER-backend [7]. SpaCy är ett öppet källkodsbibliotek för naturlig språkbehandling (NLP) utvecklat av Explosion AI [9]. Det tillhandahåller förtränade språkpipelines för ett flertal språk, däribland svenska och engelska, som innehåller flera bearbetningssteg: tokenisering (uppdelning av text i ord och meningar), ordklasstagning (POS-tagging), syntaktisk analys (dependency parsing) och namngiven entitetsigenkänning (NER).

NER-steget är det som Presidio primärt utnyttjar. SpaCy:s NER-modeller är tränade på annoterade textkorporusar och kan känna igen entitetstyper som personer (PERSON), platser (GPE) och organisationer (ORG) i löpande text. Modellernas prestanda är språkberoende. En modell tränad på svenska texter förstår svenska syntaktiska mönster och namnstrukturer, medan en engelsk modell är kalibrerad för engelska. Presidio använder spaCy:s NER-output som en signal bland flera: om spaCy identifierar ett ord som ett personnamn, och omgivande kontextord som "lösenord" eller "password" förekommer i närheten, höjer Presidio sin konfidens för att det rör sig om känslig information. Det är därför valet av spaCy-modell direkt påverkar Presidios förmåga att hantera text på ett givet målspråk [7].

Nyare forskning visar att kompakta modeller tränade för att identifiera godtyckliga entitetstyper kan prestera bättre än stora språkmodeller i zero-shot-läge (Zero-shot-läge innebär att en modell används för att lösa en uppgift utan att ha tränats eller finjusterats på exempel från just den uppgiften. Modellen förlitar sig enbart på sin förträning för att generalisera till nya, osedda entitetstyper eller domäner). GLiNER, en modell baserad på en bidirektionell transformer, överträffar både ChatGPT och finjusterade LLM-modeller på flera NER-benchmark utan att behöva finjusteras på måldata, trots att modellen är betydligt mindre [8].

GLiNER representerar en modern modellbaserad NER-metod med goda resultat på etablerade benchmark, men modellen är tränad enbart på engelskspråkig data, vilket begränsar dess direkta tillämplighet i flerspråkiga miljöer. Vidare utgör GLiNER ytterligare en variant inom den modellbaserade NER-kategorin snarare än en ny metodologisk ansats. Eftersom studiens syfte är att jämföra övergripande metodkategorier snarare än enskilda modellvarianter inkluderas den därför inte i den empiriska jämförelsen.

Microsoft Presidio kombinerar regex och NER med stöd för flera språk via utbytbara spaCy-modeller och är utformat för lokal drift, vilket gör det relevant i miljöer med höga dataskyddskrav.

Sammanfattningsvis ger den genomgångna forskningen en tydlig överblick över metodlandskapets styrkor och svagheter. Regelbaserade metoder som regex är deterministiska och resurseffektiva men begränsade till strukturerade format. NER-baserade verktyg som Microsoft Presidio kombinerar mönstermatchning med kontextuell förstärkning och stöder lokal drift. Generativa språkmodeller som Mistral 7B erbjuder kontextförståelse utan fast formatberoende och kan därmed identifiera känslig information som varken regex eller NER-baserade metoder klarar av (se avsnitt 2.4).

### 2.3 Semantisk sökning och RAG

Traditionell textsökning är syntaktisk och bygger på exakta ordmatchningar. Detta innebär att systemet identifierar dokument som innehåller samma ord som sökfrågan, men saknar förmåga att koppla samman semantiskt relaterade begrepp. En sökning på "brandvägg" returnerar exempelvis inte nödvändigtvis dokument som innehåller "firewall" eller "ACL-regler", trots att dessa behandlar närbesläktade koncept.

Semantisk sökning adresserar detta genom att representera text som numeriska vektorer, så kallade embeddings. Dessa genereras med hjälp av transformerbaserade språkmodeller som fångar kontext och betydelse snarare än enbart exakta ordmatchningar. Varje dokument och sökfråga representeras som en numerisk vektor i ett högdimensionellt rum där texter med liknande betydelse placeras nära varandra. Likheten mellan två vektorer beräknas vanligtvis med cosinuslikhet, där ett värde nära 1 indikerar hög semantisk överensstämmelse. Detta möjliggör rankning av dokument baserat på betydelse snarare än enbart exakta ordmatchningar [10].

För att möjliggöra effektiv lagring och sökning av embeddings används vektordatabaser. Dessa är optimerade för att hantera närhetsfrågor i högdimensionella rum. Exempel på open source-lösningar är FAISS, utvecklat av Meta, och Chroma, utvecklat av Chroma AI, vilka båda kan köras lokalt utan beroende av molntjänster [13][14].

Retrieval-Augmented Generation (RAG) är en arkitektur där en språkmodell kombineras med ett retrieval-steg för att generera svar baserade på externt hämtad

information. Tekniken introducerades av Lewis et al. (2021) och har visat sig förbättra faktabaserad korrekthet jämfört med modeller som enbart förlitar sig på träningsdata [11]. Senare översikter, såsom Gao et al. (2023), framhåller RAG som en etablerad metod inom moderna kunskapsbaserade system [12].

Inom informationsåtervinning kan tre huvudsakliga sökmetoder särskiljas. BM25 är en nyckelordsbaserad metod som rankar dokument utifrån termfrekvens och dokumentlängd [15]. Dense retrieval bygger istället på embeddings och semantisk likhet. Dessa metoder har komplementära styrkor: BM25 är effektiv för exakta matchningar, medan dense retrieval är bättre på att fånga betydelse och kontext. Hybridmetoder kombinerar dessa angreppssätt och beskrivs i forskning som komplementära, där nyckelordsbaserad och embeddingsbaserad sökning fångar olika relevansegenskaper och kan dra nytta av varandra [12].

## 2.4 Lokala språkmodeller och on-prem AI

Sedan 2024 har tillgängligheten av öppna och lokalt körbara språkmodeller ökat markant. Modeller såsom Llama 3 från Meta [16] och Mistral 7B [17] uppvisar hög prestanda på etablerade benchmark inom språkförståelse, resonemang och instruktionstolkning, och mindre varianter av dessa modeller kan distribueras utan dedikerad molninfrastruktur. Empiriska utvärderingar visar att modeller med 7 till 8 miljarder parametrar presterar väl på resonemang, klassificering och instruktionstolkning [17], och att öppna modeller som Llama 3 presterar i nivå med ledande slutna modeller [16], vilket gör dem användbara för automatiserad analys och kvalitetsgranskning av teknisk dokumentation.

Ollama är ett ramverk som förenklar installation och drift av lokala språkmodeller. Det erbjuder ett REST API som gör det möjligt att skicka förfrågningar till modellen precis som till en molntjänst, men utan att data lämnar den lokala miljön. Det gör Ollama till ett lämpligt val för on-prem miljöer med höga säkerhetskrav [18].

Ur ett GDPR-perspektiv är det viktigt att notera att inferens sker lokalt, men att modellens kunskapsbas är tränad på extern data. Ingen intern data skickas externt under drift, vilket är acceptabelt ur ett dataskyddsperspektiv.

## 2.5 Tidigare Arbeten

Flera forskningsområden berör delar av det system som föreslås i denna studie, däribland automatiserad anonymisering, entitetsigenkänning, RAG samt AI stöd för dokumentationssystem. Nedan presenteras centrala arbeten som relaterar till problemområdet.

### 2.5.1 Anonymisation Models for Text Data (Lison et al., 2021)

Lison et al. [5] kartlägger problemet med att automatiskt hitta och dölja känslig information i text. Författarna identifierar två huvudsakliga angreppssätt: metoder som tränar modeller att känna igen känsliga ord och fraser, samt metoder som utgår från risken att information kan avslöja en persons identitet. De argumenterar för att inget av angreppssätten ger en fullständig lösning. Genom en fallstudie jämförs tre verktyg, däribland Microsoft Presidio, och tre centrala utmaningar lyfts fram: hur

man hanterar information som bara är känslig i sitt sammanhang, hur man balanserar skydd mot användbarhet, samt hur man mäter kvaliteten på detektionen.

Studien behandlar dock inte teknisk IT-dokumentation eller miljöer där flera språk används, vilket begränsar hur direkt resultaten går att tillämpa på den kontext som detta arbete fokuserar på.

### **2.5.2 Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (Lewis et al., 2021)**

Lewis et al. introducerade en retrieval augmenterad modellarkitektur där en språkmodell kombineras med extern informationssökning. De visar empiriskt att denna kombination förbättrar prestandan på kunskapsintensiva uppgifter jämfört med traditionella sekvens till sekvens modeller. Modellen utvärderades på flera öppna frågesvarsdataset och uppvisade genomgående bättre resultat.

En viktig slutsats är att extern informationsåtervinning minskar behovet av att all kunskap måste vara lagrad i modellens parametrar och istället möjliggör mer faktabaserade och verifierbara svar.

Studien bygger dock på öppna kunskapskällor, främst Wikipedia, och behandlar inte interna dokumentationssystem eller miljöer med krav på lokal drift och dataskydd.

### **2.5.3 Analyzing the Efficacy of Microsoft Presidio in Identifying Social Security Numbers in Unstructured Text (Friebely)**

Friebely genomförde en empirisk utvärdering av Microsoft Presidios förmåga att identifiera amerikanska socialförsäkringsnummer (SSN) i ostrukturerad e-posttext [19]. Studien jämförde Presidios standardkonfiguration med en uppdaterad regex och mätte precision (andelen flaggade entiteter som faktiskt var känsliga), recall (andelen känsliga entiteter som hittades) och F1-score (medelvärde av de två) samma mätvärden som används i detta arbete.

Resultaten visade att Presidio i standardkonfiguration presterade sämre än efter att regex-biblioteket uppdaterats. En central slutsats var att Presidio kräver domänspecifik konfiguration för att prestera optimalt, något som även bekräftas i detta arbete där Presidio utan svensk spaCy-modell presterade sämre än ren regex.

Studien är dock begränsad till engelskspråkig text och ett enda entitetsformat, vilket innebär att dess resultat inte är direkt överförbara till svenska tekniska texter med en bredare uppsättning entitetstyper. Denna begränsning motiverar den egna utvärderingen i detta arbete, där Presidio konfigureras och testas specifikt för svenska texter med sex entitetskategorier.



## 3 Metod

Detta kapitel beskriver de metoder och system som implementerades för att besvara studiens två frågeställningar. Vidare redogörs för hur testdatat konstruerades, hur utvärderingen genomfördes samt vilken hårdvara och mjukvara som användes under experimenten.

### 3.1 Metoder för GDPR-detektion

#### 3.1.1 Översikt

För att besvara den första frågeställningen (F1), hur väl en språkmodell identifierar GDPR-känslig information jämfört med regelbaserade metoder, implementerades och utvärderades tre detektionsmetoder: regex, Microsoft Presidio och Mistral 7B via Mistral API. Metoderna representerar tre huvudsakliga metodkategorier för identifiering av känslig information: regelbaserade metoder, NER-baserade metoder och generativa språkmodeller. Samtliga metoder testades på samma syntetiska testset och utvärderades med Precision, Recall och F1-score mot ett manuellt annoterat facit. Alla metoder är dessutom körbara i on-prem-miljö, vilket är ett krav givet Axians dataskyddspolicy.

#### 3.1.2 Testset

Testdatat består av tolv syntetiskt genererade dokument som efterliknar teknisk IT-dokumentation av den typ som förekommer i Axians FosWiki-miljö. Sex dokument är skrivna på svenska och sex på engelska, vilket speglar den faktiska tvåspråkiga karaktären hos Axians interna dokumentation och möjliggör en bedömning av metodernas språkoberoende. Fördelningen av entitetstyper framgår av tabell 3.1.

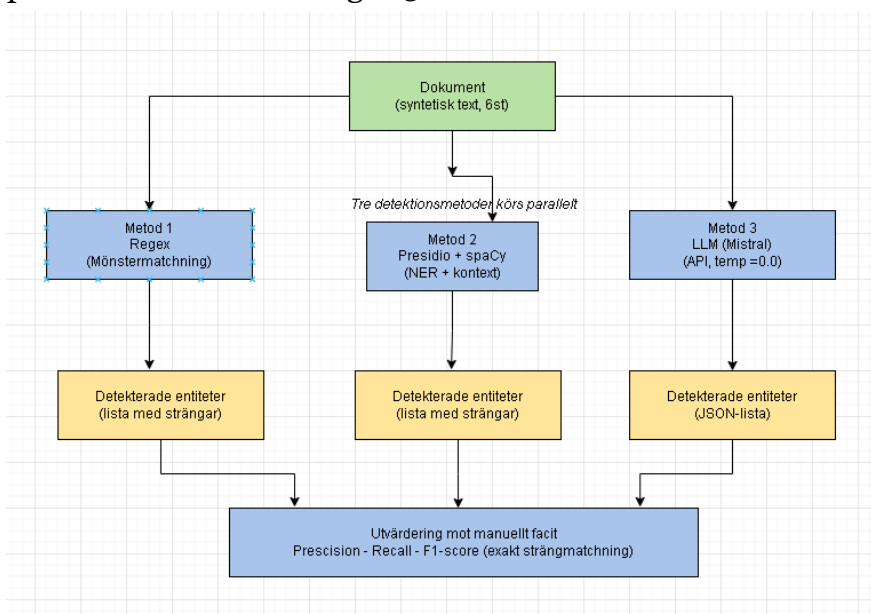
Dokumenterna innehåller totalt 30 annoterade entiteter av typerna IP-adress, lösenord, API-nyckel, personnamn, e-postadress, personnummer och telefonnummer. Två av dokumenten, ett på svenska och ett på engelska, innehåller ingen känslig information och inkluderades för att testa metodernas benägenhet att generera falska larm.

Testdatat är syntetiskt för att undvika att faktisk känslig information från Axians hanteras utanför organisationens nätverk. De engelska dokumenten konstruerades med samma entitetstyper och strukturella mönster som de svenska, men med engelska namn, domäner och meningsuppbyggnad för att testa metodernas förmåga att hantera en tvåspråkig dokumentationsmiljö.

Tabell 3.1: Fördelning av entitetstyper i det utökade testdatat.

Entitetstyp	Svenska dokument	Engelska dokument	Totalt
IP-adress	4	4	8
Lösenord	4	4	8
E-postadress	2	3	5
Personnamn	2	2	4
API-nyckel	1	1	2
Personnummer	1	1	2
Telefonnummer	1	1	2
<b>Totalt</b>	<b>15</b>	<b>16</b>	<b>30</b>

Testsetets storlek, 30 entiteter i 12 dokument är en medveten avgränsning. Det är tillräckligt för att demonstrera och jämföra metodernas karaktäristiska beteenden, exempelvis om en metod systematiskt missar kontextberoende entiteter eller uppvisar sämre prestanda på ett av de två språken, men för litet för att dra statistiskt generaliserbara slutsatser om precisionstal. Resultaten ska därför tolkas som indikatorer på metodernas principiella styrkor och svagheter snarare än som definitiva prestandavärden. En bredare utvärdering med fler dokument identifieras som ett område för fortsatt arbete. Hur testningen av dessa dokument går till i praktiken illustreras i Figur 3.1.



Figur 3.1: Flödesschema för GDPR-detektionssystemet. Varje testdokument skickas parallellt till de tre detektionsmetoderna och utvärderas mot manuellt annoterat facit.

### 3.1.3 Visuellt exempel på testdata

Figur 3.2 visar ett exempel på ett av de manuellt konstruerade testdokumenten tillsammans med dess annoteringar. Texten är utformad för att efterlikna en teknisk driftmiljö och innehåller avsiktligt inbäddad känslig information i löpande instruktionstext. Exemplet illustrerar hur en IP-adress, ett lösenord, ett personnamn och en e-postadress förekommer naturligt i texten och samtidigt är strukturerat annoterade med respektive entitetstyp IP\_ADDRESS, PASSWORD, PERSON och EMAIL. Figuren demonstrerar därmed hur testdokumenten är uppbyggda samt hur facit används för att möjliggöra kvantitativ utvärdering av olika detektionsmetoder.

```
"Servern för produktionsmiljön nås på IP-adress 192.168.1.45. "
"Administratörskontot heter admin och lösenordet är Axians2024!. "
"Kontakta Erik Svensson på erik.svensson@axians.se vid problem.",
"annotations": [
  {"text": "192.168.1.45", "type": "IP_ADDRESS"},
  {"text": "Axians2024!", "type": "PASSWORD"},
  {"text": "Erik Svensson", "type": "PERSON"},
  {"text": "erik.svensson@axians.se", "type": "EMAIL"}
]
```

Figur 3.2: Exempel på manuellt konstruerat testdokument med tillhörande annoteringar av känslig information.

### 3.1.4 Metod 1 - Regex

Den regelbaserade metoden använder ett antal reguljära uttryck som matchar vanliga format för känslig information. Mönster definierades för IP-adresser, e-postadresser, telefonnummer, personnummer. Metoden är deterministisk och ger alltid samma utdata för samma indata. Den har inget beroende av externa modeller och kräver inga beräkningsresurser utöver det som behövs för mönstermatchning.

En grundläggande begränsning är att metoden enbart kan identifiera information med ett fast, fördefinierat format, vilket innebär att kontextberoende känslig information inte kan detekteras (se avsnitt 2.2).

### 3.1.5 Metod 2 - Microsoft Presidio

Som beskrivs i avsnitt 2.2 är valet av spaCy-modell avgörande för Presidios förmåga att hantera ett givet målspråk. I det här arbetet konfigurerades Presidio med både den svenska modellen sv\_core\_news\_lg och den engelska modellen en\_core\_web\_lg, med automatisk språkdetektering per dokument via biblioteket langdetect. Detta möjliggör att kontextordsförstärkningen är kalibrerad för respektive språk, svenska kontextord som "lösenord" och "användare" hanteras av den svenska modellen, medan engelska motsvarigheter som "password" och "address" hanteras av den engelska. Utan språkspecifika modeller baseras NER-modellens kontextuella förstärkning enbart på det träningspråk som modellen är kalibrerad för, vilket försämrar detektionen på det andra språket. Konfigurationen med dubbla modeller valdes därför för att spegla den faktiska tvåspråkiga karaktären hos Axians

FosWiki-miljö och möjliggöra en rättvis jämförelse på båda delmängderna av testdatat.

### 3.1.6 Metod 3 - LLM via Mistral API

Den tredje metoden använder Mistral 7B via ett HTTP-anrop till Mistral API. Modellen instrueras via ett prompt att identifiera känslig information och returnera resultatet som en JSON-lista. LLM-resultaten cachas efter första körningen för att undvika att modellen anropas flera gånger på samma dokument.

Det är viktigt att notera att detta test körs via ett molnbaserat API, vilket i produktionsmiljö hos Axians inte är tillåtet eftersom intern data inte får lämna organisationens nätverk. Den modell som utvärderas via API motsvarar den modell som är avsedd att implementeras lokalt. I en lokal implementation ersätts API-anropet med ett internt Ollama-anrop till samma modell.

Mistral 7B är en instruktionsföljande modell utan språkbegränsning, vilket innebär att den hanterar både svenska och engelska dokument utan ytterligare konfiguration. Detta är en fördel gentemot Presidio, som kräver språkspecifika spaCy-modeller för optimal prestanda.

### 3.1.7 Utvärderingsmetodik

Varje metod utvärderades mot samma facit med hjälp av exakt strängmatchning, där en detektion räknas som korrekt om den detekterade texten exakt matchar en annoterad entitet. Tre mätvärden används för att utvärdera detektionsmetoderna: Precision, Recall och F1-score. Dessa valdes eftersom de är etablerade standardmått inom klassificering och informationsåtervinning för uppgifter där både falska positiva och falska negativa är relevanta.

Precision mäter hur tillförlitlig en metod är när den flaggar något som känsligt det vill säga andelen av alla flaggade entiteter som faktiskt är känsliga. En metod med hög precision genererar få falska larm. Recall mäter hur heltäckande metoden är andelen av all känslig information i testdatat som faktiskt identifierades. En metod med hög recall missar få entiteter. De två måtten står ofta i konflikt: en metod som flaggar allt får perfekt recall men låg precision, medan en metod som bara flaggar när den är helt säker får hög precision men potentiellt låg recall.

F1-score är det harmoniska medelvärdet av precision och recall och kombinerar de två till ett enda jämförbart mätvärde. Till skillnad från ett vanligt medelvärde straffar F1-score metoder som presterar extremt olika på de två dimensionerna; en metod med perfekt recall men mycket låg precision får därmed ett lågt F1-score. Formellt beräknas F1 som [22]:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (1)$$

Alternativa mått såsom accuracy bedömdes mindre lämpliga eftersom datasetet innehåller betydligt fler negativa än positiva textsegment. Ett system skulle därför kunna uppnå hög accuracy genom att klassificera majoriteten av texten som icke-känslig, trots att viktiga entiteter missas. Precision, Recall och F1-score ger därför en

mer rättvisande utvärdering av detektionsförmågan i denna typ av obalanserad klassificeringsuppgift.

I kontexten av GDPR-detektion är recall särskilt viktigt, eftersom en missad känslig entitet innebär en potentiell säkerhets- eller integritetsbrist. Precision är dock också relevant, eftersom ett system som genererar för många falska larm skapar onödig manuell granskningsbörda [5].

Utvärderingen genomfördes över samtliga dokument samt separat för de svenska respektive engelska delmängderna, för att synliggöra eventuella språkberoende skillnader i metodernas prestanda.

En begränsning i utvärderingen är att exakt strängmatchning kan underskatta LLM-prestandan. Om modellen extraherar en delvis korrekt sträng räknas det som både ett miss och ett falskt larm trots att modellen identifierade rätt känslig information.

### 3.1.8 Implementationsdetaljer

Systemet implementerades i Python 3.13. Regex-mönstren definierades manuellt baserat på domänkunskap om vanliga format för känslig IT-information. Presidio konfigurerades med `sv_core_news_lg` för svenska dokument och `en_core_web_lg` för engelska dokument, med automatisk språkdetektering via `langdetect` som dirigerar varje dokument till rätt analysator. Mistral-anropet skickades via `requests`-biblioteket och ett strukturerat systemprompt som specificerade de entitetstyper som skulle identifieras.

Alla tre metoder kördes i samma miljö mot identiska dokument. Evalueringslogiken beräknar precision, recall och F1-score per entitetstyp och aggregerat, vilket möjliggör en detaljerad analys av var varje metod lyckas respektive misslyckas.

## 3.2 Sökmeter

### 3.2.1 Översikt

För att besvara studiens andra frågeställning (F2), hur mycket sökkvaliteten förbättras vid användning av semantisk sökning jämfört med syntaktisk sökning, implementerades och jämfördes tre söksystem: (1) en syntaktisk baslinjemetod, (2) ett semantiskt embeddingsbaserat system samt (3) en hybridmetod som kombinerar resultaten från de två tidigare systemen.

Valet av metoder motiveras av att de representerar tre centrala angreppssätt inom informationsåtervinning och möjliggör en systematisk jämförelse mellan befintlig sökfunktionalitet och mer avancerade AI-baserade metoder. Den syntaktiska metoden fungerar som baslinje eftersom den motsvarar den typ av sökning som används i FosWiki, vilket gör det möjligt att kvantifiera förbättringar i relation till nuvarande system.

Samtliga system utvärderas på samma dataset och med identiska sökfrågor, vilket säkerställer en rättvis och reproducerbar jämförelse.

### 3.2.2 Testset

För utvärdering av söksystemen konstruerades ett annoterat testset bestående av 20 sökfrågor kopplade till en syntetisk dokumentkorpus som efterliknar teknisk IT-dokumentation. Dokumenten är skrivna på både svenska och engelska för att spegla den tvåspråkiga karaktären hos Axians FosWiki-miljö.

Frågorna är utformade för att representera olika typer av sökscenarier som är relevanta i praktiken, inklusive synonymbaserade frågor, parafraaser, konceptuella frågor, tvärspråkiga frågor samt ett kontrollfall med exakt terminologimatchning. Denna variation möjliggör en systematisk analys av hur olika sökmetoder hanterar både exakta matchningar och semantiskt relaterade uttryck.

Varje fråga är kopplad till 3–4 dokument med relevansnivåer från 0 till 3, där 0 indikerar ingen relevans och 3 indikerar hög relevans. Den graderade relevansskalan möjliggör en mer nyanserad utvärdering jämfört med binär klassificering.

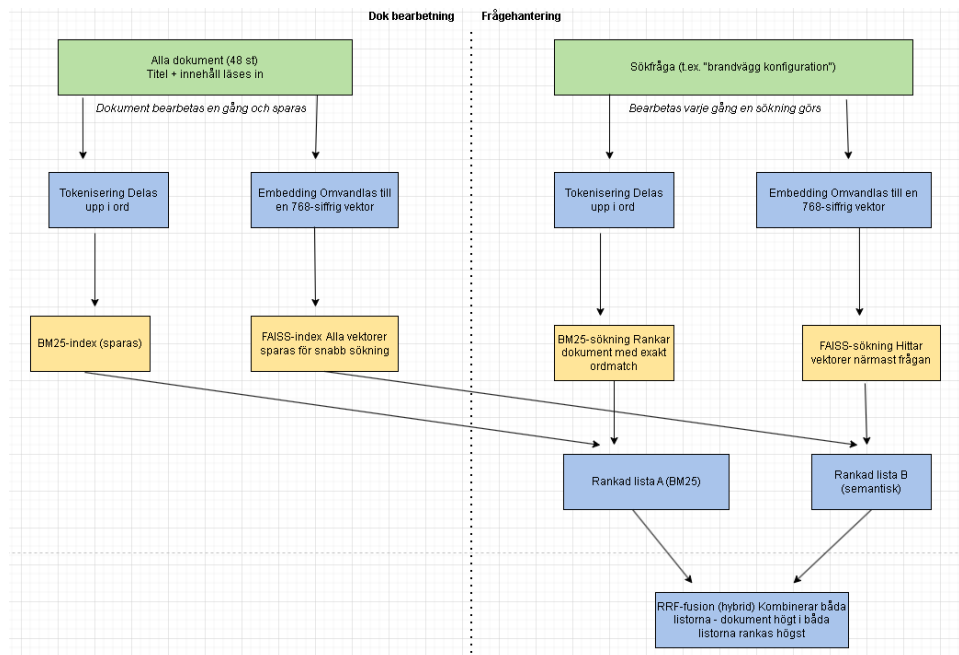
Testsetet innehåller totalt 48 unika dokument. Samtliga dokument indexeras av alla tre söksystem för att säkerställa jämförbarhet mellan metoderna.

Frågorna är fördelade på sex frågetyper som är utformade för att täcka de centrala utmaningarna med syntaktisk sökning i tvåspråkig teknisk dokumentation. Fördelningen framgår av tabell 3.2.

Tabell 3.2: Fördelning av frågetyper i testsetet för semantisk sökning.

Frågetyp	Antal	Beskrivning	Exempel fråga
synonym_sv_en	7	Svensk fråga, engelska dokument	brandvägg konfiguration
synonym_en_sv	4	Engelsk fråga, svenska dokument	password policy requirements
Paraphrase	5	Naturlig fråga utan exakta sökord	hur återställer jag ett raderat konto
Conceptual	3	Kräver teknisk konceptförståelse	vilka portar behöver öppnas för RDP
cross_lingual	1	Explicit tvärspråkig sökning	network monitoring alerts
exact_match	1	Kontrollfall, exakt terminologi	kubernetes container orchestration
<b>Totalt</b>	<b>20</b>		

För att ge en översiktlig bild av systemets arkitektur illustreras det övergripande flödet i Figur 3.3. Den vänstra delen av diagrammet visar indexeringsfasen där dokumenten förbereds genom tokenisering och vektorisering. Den högra delen visar sökfasen där frågor bearbetas genom parallella sökningar som sedan sammanställs med hjälp av Reciprocal Rank Fusion (RRF).



Figur 3.3: Övergripande flödesschema för söksystemets arkitektur.

### 3.2.3 System A - Syntaktisk sökning med BM25

Den syntaktiska baslinjemetoden implementeras med BM25 via biblioteket rank-bm25. Dokumenten tokeniseras med en regexbaserad tokeniserare, normaliseras till gemener och delas upp i ord baserat på blanksteg och skiljetecken.

BM25 valdes som baslinje eftersom det representerar den typ av nyckelordsbaserad sökning som används i befintliga wikisystem. Detta möjliggör en direkt jämförelse mellan nuvarande sökbeteende och mer avancerade metoder. Ingen semantisk normalisering eller synonymhantering tillämpas, vilket innebär att endast exakta ordmatchningar påverkar rankingen.

### 3.2.4 System B - Semantisk sökning med sentence-transformers och FAISS

Det semantiska systemet baseras på embeddingmodellen paraphrase-multilingual-mpnet-base-v2 från sentence-transformers. Modellen valdes för sin förmåga att hantera både svenska och engelska texter i samma representationsrum, vilket är centralt för den aktuella dokumentmiljön.

Samtliga dokument omvandlas till vektorer och indexeras i FAISS. För att möjliggöra effektiv likhetsberäkning normaliseras vektorerna till enhetlig längd.

Vid sökning omvandlas frågan till en embedding med samma modell, och de  $k$  närmaste dokumentvektorer identifieras baserat på cosinuslikhet. Detta möjliggör matchning baserat på semantisk likhet snarare än exakta ord.

### 3.2.5 System C - Hybridmetod med Reciprocal Rank Fusion

Hybridmetoden kombinerar resultaten från BM25 och det semantiska systemet med hjälp av Reciprocal Rank Fusion (RRF). För varje dokument beräknas ett kombinerat poäng baserat på dess position i respektive rankad lista enligt:

$$score = \sum \frac{1}{k+rank} \quad (2)$$

där  $k = 60$  är en konstant enligt standardinställningar i tidigare forskning [20].

RRF valdes eftersom metoden inte kräver parameteroptimering och är robust mot skillnader i poängskalor mellan de underliggande systemen. Detta gör den särskilt lämplig i en experimentell kontext där fokus ligger på jämförelse snarare än optimering.

Syftet med hybridmetoden är att undersöka om en kombination av syntaktisk och semantisk sökning kan utnyttja båda metodernas styrkor och därigenom förbättra sökprestandan.

### 3.2.6 Utvärderingsmetodik

Alla tre söksystem utvärderas mot samma annoterade testset med tre standardmätt inom informationsåtervinning: Precision@ $k$ , Recall@ $k$  och MRR.

Precision@ $k$  mäter hur träffsäker sökningen är bland de  $k$  översta resultaten, det vill säga andelen av de  $k$  returnerade dokumenten som faktiskt är relevanta. Ett högt P@1 innebär att det allra första resultatet nästan alltid är relevant, vilket är avgörande för användbarhet i praktiken. Recall@ $k$  mäter hur heltäckande sökningen är bland de  $k$  översta resultaten, andelen av alla relevanta dokument i korpusen som systemet lyckas returnera inom topp- $k$ . Ett högt R@10 innebär att systemet hittar de flesta relevanta dokumenten om användaren tittar på de tio första träffarna.

MRR, Mean Reciprocal Rank, fokuserar på var det första relevanta dokumentet dyker upp i rankingen. För varje fråga beräknas det reciproka rankningsvärdet som  $1/rank$ , där rank är positionen för det första relevanta dokumentet. Om det relevanta dokumentet är på plats 1 ger det värdet 1,0; på plats 2 ger det 0,5; på plats 3 ger det 0,33, och så vidare. MRR är sedan medelvärdet av dessa värden över alla frågor. Ett MRR nära 1,0 innebär att det första relevanta dokumentet konsekvent hamnar högt i resultatlistan [21].

Mätvärden beräknas för  $k \in \{1, 3, 5, 10\}$ , vilket möjliggör en analys av hur systemen presterar både vid smala sökningar ( $k=1$ ) och bredare informationssökning ( $k=10$ ). Relevansströskeln är satt till 1, vilket innebär att dokument med relevansnivå 1, 2 eller 3 räknas som relevanta.

Mätvärdesberäkningarna implementerades från grunden i Python, utan externa utvärderingsbibliotek, för transparens och reproducerbarhet.

### 3.2.7 Implementationsdetaljer

Notebooken implementerades i Python 3.13 med biblioteken rank bm25 version 0.2.2, sentence transformers version 5.3.0 och faiss cpu version 1.13.2. Testsettet lagras som JSON med strukturerade relevansnivåer per dokument och fråga. Samtliga 48 unika dokument i testsettet indexeras av alla tre system. Ground truth byggs som en nyckel värde tabell med {query\_id: {doc\_id: relevance}} för direkt jämförelse mot de rankade resultaten.

Resultat exporteras som CSV filer per fråga och aggregerat. Notebookens kod är reproducerbar och körbar utan externt API beroende för BM25 och semantisk sökning. LLM komponenten i GDPR detektion kräver Mistral API nyckel eller en lokal Ollama installation.

## 3.3 Testmiljö och hårdvara

Alla experiment genomfördes på en bärbar dator med följande hårdvara, specificerat i tabell 3.3:

Tabell 3.3: Hårdvara och mjukvarumiljö för experimentkörningarna.

Komponent	Specifikation
Processor	Apple M2 (8 kärnor, unified memory)
Arbetsminne (RAM)	8 GB unified memory
Operativsystem	macOS
Python-version	Python 3.13 (Miniconda-miljö)
Beräkningsenhet	CPU/Neural Engine. Inga separata GPU-bibliotek (faiss-cpu, PyTorch CPU-backend)
Embeddingsgenerering	Ca 2 s för 48 dokument (mpnet, 2 batchar)
Mistral API	Molnbaserat (Mistral 7B). Kördes mot manuellt skapade testdokument utan faktisk känslig information

Apple M2 är en ARM baserad SoC, System on Chip, med unified memory arkitektur, vilket innebär att CPU och Neural Engine delar samma minnesutrymme. Eftersom faiss cpu och PyTorch CPU backend användes krävdes ingen separat GPU, vilket är

representativt för en on-prem miljö utan dedikerad AI accelerator. Embeddingsgenerering för 48 dokument tog cirka 2 sekunder, vilket är acceptabelt för en prototyp. I en produktionsmiljö med ett större dokumentindex kan indexeringstiden öka linjärt, men söktiden per fråga förblir konstant tack vare FAISS IndexFlatIP strukturen.





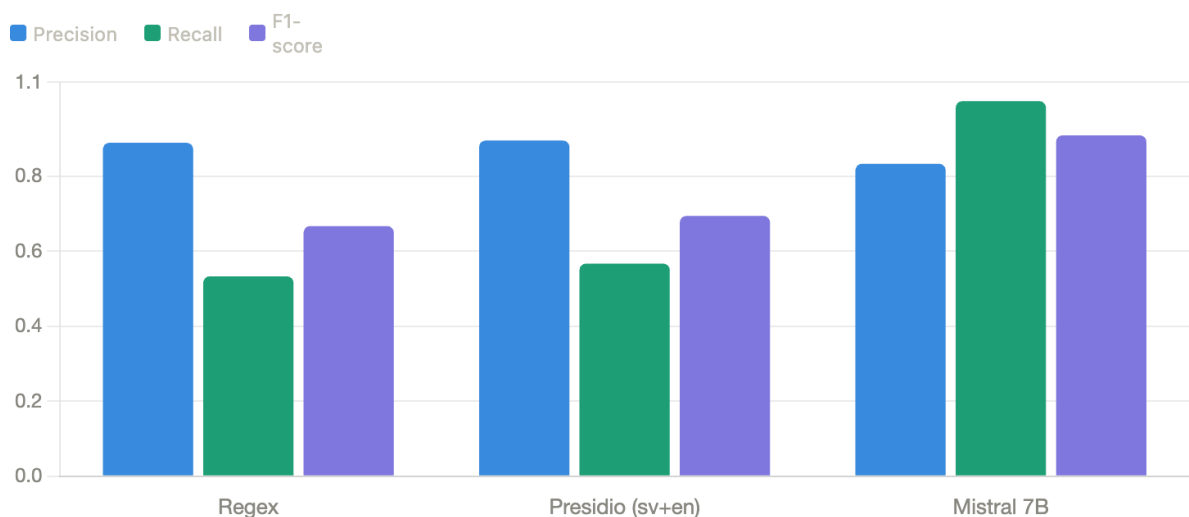
## 4 Resultat

Detta kapitel presenterar resultaten från de genomförda experimenten, uppdelat på GDPR-detektion och informationsåtervinning, och redovisar mätvärden både aggregerat och per underkategori.

### 4.1 GDPR-detektion

#### 4.1.1 Aggregerade mätvärden

Figur 4.1 visar de aggregerade mätvärdena för de tre detektionsmetoderna på testdatat med 30 annoterade entiteter i tolv syntetiskt skapade dokument. LLM-metoden uppnår det högsta F1-score på 0,909, drivet av en perfekt recall på 1,000 och samtliga 30 annoterade entiteter identifierades. Precision på 0,833 indikerar fem falska larm över samtliga tolv dokument. Regex uppnår F1 = 0,667 med hög precision 0,889 men låg recall 0,533. Presidio med svenska och engelska spaCy-modeller uppnår F1 = 0,694 med precision 0,895 och recall 0,567. Fullständiga resultat per entitetstyp och dokument redovisas i Bilaga A.



Figur 4.1 visar de aggregerade mätvärdena för de tre detektionsmetoderna på testdatat med 30 annoterade entiteter i tolv syntetiskt skapade dokument, sex på svenska och sex på engelska.

### 4.1.2 Prestanda per entitetstyp

Mönstret är konsistent med teorin: Regex och Presidio hittar entiteter med fast format, IP-adresser, personnummer och e-postadresser med standard-TLD, men missar helt entiteter som kräver kontextförståelse. Lösenord utan fast format, samtliga personnamn samt API-nycklarna hittades av varken Regex eller Presidio. Mistral identifierade samtliga entiteter i alla sju kategorier på båda språken. Fullständiga data redovisas i Bilaga A, tabell A.2.

### 4.1.3 Resultat per dokument

Detaljerade resultat per dokument för samtliga tre metoder redovisas i Bilaga A, tabellerna A.3, A.4 och A.5, vilka visar antalet korrekta träffar (TP), falska larm (FP) och missar (FN) per metod och dokument. Dokument 5 och 11 innehåller ingen känslig information och inkluderades för att testa benägenheten att generera falska larm. Samtliga tre metoder undviker korrekt att flagga dessa.

### 4.1.4 Falska positiva och falska negativa

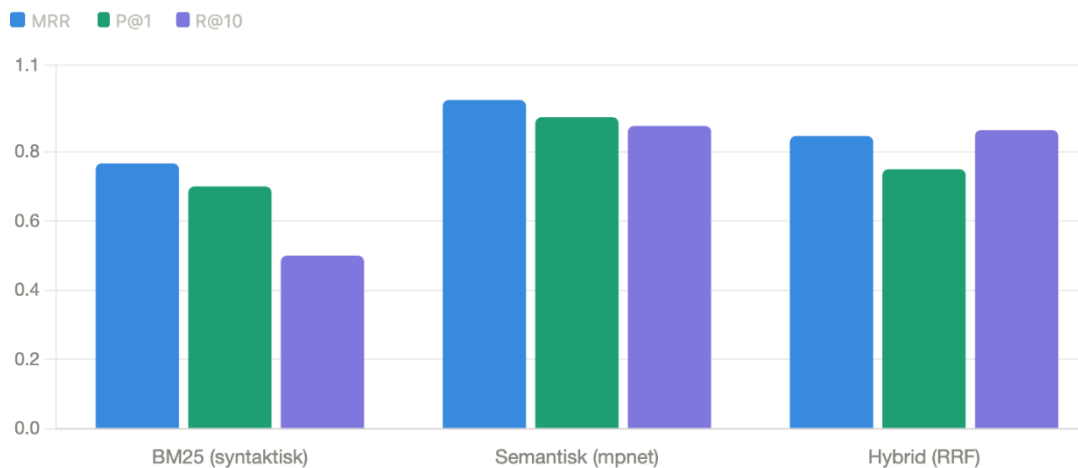
En fullständig sammanställning av samtliga falska positiva och falska negativa per metod redovisas i Bilaga A, tabell A.6. Nedan följer en analys av de viktigaste felfallen.

Regex genererade två falska larm: datum i formatet ÅÅÅÅ-MM-DD matchades av personnummERMönstret  $\{6\}[-s]?\{4\}$ . Presidio genererade ett falskt larm där "systemägaren" klassificerades som PERSON. Mistral genererade fem falska larm, inklusive systemkonton (admin, dbadmin, sysadmin), hostnamnen vpn.axians.internal samt filvägen /home/deploy/.ssh/id\_rsa.

## 4.2 Semantisk sökning

### 4.2.1 Aggregerade mätvärden

Figur 4.2 visar de aggregerade mätvärdena för de tre söksystemen över samtliga 20 frågor. Det semantiska systemet presterar bäst på samtliga mätvärden med MRR = 0,950, vilket innebär att det första relevanta dokumentet i genomsnitt hamnar på position ett. BM25 uppnår MRR = 0,767 och hybridmetoden MRR = 0,846. Recall vid  $k = 10$  är god för samtliga system, men det semantiska systemet och hybridmetoden är tydligt överlägsna BM25. Fullständiga mätvärden per frågetyp redovisas i Bilaga B.



Figur 4.2: Aggregerade mätvärden för söksystemen

#### 4.2.2 Resultat per frågetyp

Skillnaden mellan systemen är störst på synonymfrågor. Frågan 'brandvägg konfiguration' illustrerar detta tydligt. Eftersom BM25 enbart matchar på exakta ord returnerar systemet dokument som innehåller ordet 'brandvägg' men som handlar om helt andra ämnen, exempelvis VPN, DNS och PKI. Inga av de relevanta brandväggsdokumenten finns med i topp-5, vilket ger MRR = 0 för denna fråga. Det semantiska systemet förstår däremot att frågan handlar om brandväggskonfiguration även om exakt samma ord inte används i dokumenten, och returnerar samtliga fyra mest relevanta brandväggsdokument i topp-5, vilket ger MRR = 1,0.

På kontrollfallet med exakt terminologi, där sökfrågan innehåller precis samma ord som finns i dokumenten, uppnår både BM25 och det semantiska systemet MRR = 1,0. Detta bekräftar att det semantiska systemet inte tappar prestanda på exakta matchningar, utan presterar lika bra som BM25 även i de fall där nyckelordsbaserad sökning traditionellt är som starkast.

Hybridmetodens MRR hamnar konsekvent mellan BM25 och det semantiska systemet. En trolig förklaring är att hybridmetoden kombinerar de två systemens rankningar med lika vikt, vilket innebär att BM25:s svaga resultat på synonymfrågor drar ned det sammanslagna resultatet snarare än att de två systemen kompletterar varandra.



## 5 Analys och diskussion

Detta kapitel analyserar resultaten från de två experimentdelarna i relation till studiens frågeställningar och den teoretiska grunden, diskuterar metodernas styrkor och begränsningar samt sätter arbetet i ett bredare samhälleligt sammanhang.

### 5.1 Tolkning av GDPR-detektionsresultaten

Det övergripande resultatet bekräftar den teoretiska bild som presenterades i kapitel 2: metodernas prestanda är direkt kopplad till om den känsliga informationen följer ett fast format eller kräver kontextförståelse. Regex och Presidio uppnådde recall på 0,533 respektive 0,567, vilket innebär att nästan hälften av all känslig information i testdatat missades. Med andra ord identifierades färre än sex av tio PII-förekomster, vilket i ett verkligt system skulle lämna ett betydande antal känsliga uppgifter obemärkta. I ett GDPR-perspektiv är detta problematiskt; en missad entitet utgör en potentiell säkerhets- eller integritetsbrist som traditionella verktyg inte kan fånga upp.

LLM-metoden (Mistral 7B) uppnådde perfekt recall på 1,000 och F1-score på 0,909, vilket innebär att modellen hittade samtliga 30 annoterade entiteter i testdatat utan att missa något, samtidigt som andelen falska larm hölls relativt låg. Detta är ett tydligt bättre resultat. Att modellen identifierade samtliga entiteter på båda språken utan ytterligare konfiguration bekräftar att generativa språkmodeller har en fundamental fördel i kontextberoende scenarion. En mening som "logga in med lösenordet Hej2024!" innehåller inget fast mönster att matcha mot, men är trivial för en språkmodell att tolka.

De fem falska larmen från Mistral är värda att analysera närmare. Tre av dem, nämligen admin, dbadmin och sysadmin, flaggades som personnamn. Detta är inte nödvändigtvis felaktigt ur ett säkerhetsperspektiv: systemkonton med administratörsbehörighet är känsliga identifierare även om de inte är personuppgifter i GDPR:s mening. I praktiken kan sådana konton ge obehörig åtkomst till känsliga system om de exponeras. Detsamma gäller filvägen /home/deploy/.ssh/id\_rsa, som pekar direkt mot en privat SSH-nyckel. Att modellen flaggar dessa kan ses som ett tecken på att den har en bredare säkerhetsmedvetenhet än vad det annoterade facit mäter, snarare än ett renodlat fel.

En metodologisk begränsning är att exakt strängmatchning användes som utvärderingsmetod. Om Mistral extraherar en semantiskt korrekt men textuellt något avvikande sträng räknas det som både ett miss och ett falskt larm, vilket missgynnar LLM-metoden. En mer nyanserad utvärdering med partiell matchning eller manuell granskning skulle sannolikt ge Mistral ett ännu högre F1-score.

### 5.2 Tolkning av sökresultaten

Det semantiska systemet (mpnet) uppnådde MRR = 0,950 jämfört med BM25:s 0,767, en förbättring som är konsistent över samtliga frågetyper. Skillnaden är störst på synonymfrågor, vilket är precis det scenario som motiverade frågeställningen. Frågan "brandvägg konfiguration" returnerade inga relevanta resultat med BM25,

medan det semantiska systemet placerade samtliga fyra relevanta dokument i topp-5. Detta illustrerar tydligt varför syntaktisk sökning är otillräcklig i en tvåspråkig teknisk dokumentationsmiljö där samma koncept uttrycks på olika sätt.

Att hybridmetoden (RRF,  $MRR = 0,846$ ) inte överträffade det semantiska systemet ensamt var det mest oväntade resultatet. En trolig förklaring är att RRF behandlar de underliggande systemens rankningar lika oavsett relativ styrka. På synonymfrågor är BM25-rankningen så svag att den drar ned fusionsresultaten snarare än att komplettera dem. RRF är robust men okänslig för hur stor prestandaskillnaden är mellan de underliggande systemen. En viktad fusion, där det semantiska systemet ges högre vikt på frågor utan exakta ordmatchningar, hade sannolikt gett bättre resultat. Detta identifieras som ett område för fortsatt arbete.

Kontrollfallet "kubernetes container orchestration" bekräftar att det semantiska systemet inte tappar prestanda på exakta matchningar, båda systemen uppnår  $MRR = 1,0$ . Det semantiska systemet är alltså inte ett val man gör på bekostnad av precision vid exakta sökningar, utan ett komplement som tillför värde utan att försämra det som redan fungerar.

### 5.3 Alternativa lösningar

GLiNER övervägdes som ett alternativ till Presidio men exkluderades eftersom det representerar samma metodologiska kategori, modellbaserad NER snarare än en ny ansats. Studiens syfte var att jämföra övergripande metodkategorier, och en jämförelse mellan två NER-varianter hade inte besvarat frågeställningen på ett mer principiellt sätt.

En fullständig RAG-pipeline med genererande LLM implementerades inte, vilket är en medveten avgränsning motiverad av tidsramen och att sökning och generering är separata steg med olika utvärderingskrav. En fullständig RAG-implementation, där det semantiska söksystemet kopplas till en lokal Ollama-instans för svarsgenerering, identifieras som ett naturligt nästa steg för fortsatt arbete.

### 5.4 Ekonomiska, sociala, etiska och miljömässiga aspekter

Ur ett ekonomiskt och samhällligt perspektiv är automatiserad GDPR-detektion i interna dokumentationssystem relevant för alla IT-organisationer som hanterar känslig infrastrukturinformation. Kostnaden för en GDPR-överträdelse kan vara betydande, och ett system som tidigt identifierar känslig information i dokumentation minskar risken för oavsiktliga dataintrång. Den on-prem-orienterade ansatsen är särskilt relevant för organisationer inom offentlig sektor och kritisk infrastruktur där molntjänster inte är ett alternativ av säkerhetsskäl.

Ur ett etiskt perspektiv är det viktigt att ett automatiserat detektionssystem inte ersätter mänsklig granskning utan fungerar som ett beslutsstöd. Resultaten visar att LLM-metoden ger perfekt recall men med ett antal falska larm som kräver manuell uppföljning. Ett system som flaggar för mycket riskerar att förlora användarnas förtroende, medan ett system som missar för mycket ger en falsk trygghet. Balansen mellan dessa två risker måste kommuniceras tydligt till de som använder systemet.

Den miljömässiga aspekten kopplas till valet av teknisk arkitektur. Mistral 7B kräver mer beräkningsresurser än regex eller Presidio, men körs i detta sammanhang på befintlig serverhårdvara och utan kontinuerlig drift, inferens sker vid behov och inte i realtid. Energikostnaden är därmed begränsad i förhållande till nyttan. Användningen av lokala modeller via Ollama eliminerar dessutom behovet av att skicka data till externa molntjänster, vilket minskar nätverksberoende och ger bättre kontroll över dataflöden.



## 6 Slutsatser

Detta kapitel sammanfattar de viktigaste resultaten och insikterna från arbetet, ger en övergripande bedömning av hur väl målsättningarna har uppnåtts samt identifierar lärdomar och områden för fortsatt arbete.

Studiens två frågeställningar kan nu besvaras som följer:

*F1: Hur väl identifierar en språkmodell GDPR-känslig information i teknisk text jämfört med regelbaserade metoder?*

I det genomförda testsetet identifierade Mistral 7B känslig information markant bättre än regelbaserade metoder. Modellen uppnår perfekt recall och F1-score på 0,909, medan regex och Presidio uppnår recall under 0,57 och missar systematiskt kontextberoende entiteter som lösenord, personnamn och API-nycklar. Kontextförståelse är avgörande, och regelbaserade metoder har begränsad förmåga att hantera denna typ av kontextberoende information.

*F2: Hur mycket förbättras sökkvaliteten när semantisk sökning används istället för syntaktisk sökning?*

I det utvärderade testsetet förbättrades sökkvaliteten påtagligt. Det semantiska systemet uppnår  $MRR = 0,950$  jämfört med BM25:s 0,767, med störst förbättring på synonymfrågor och tvärspråkliga sökningar. Viktigt är också att förbättringen sker utan prestationsförlust på exakta matchningar, vilket gör semantisk sökning till ett överlägset alternativ utan avvägningar.

### 6.1 Arbetets bidrag och resultat

För GDPR-detektion visar studien att en generativ språkmodell (Mistral 7B) identifierar samtliga typer av känslig information i teknisk text, inklusive kontextberoende entiteter som lösenord och personnamn, med ett F1-score på 0,909 och perfekt recall. Regelbaserade metoder som regex och Microsoft Presidio uppnår recall under 0,57 och missar systematiskt entiteter utan fast format. Resultatet bekräftar att kontextförståelse är avgörande för heltäckande GDPR-detektion och att regelbaserade metoder, trots sin enkelhet och deterministiska natur, inte är tillräckliga i miljöer där känslig information förekommer i löpande text utan ett förutsägbart format.

För semantisk sökning uppnår det embeddingsbaserade systemet (paraphrase-multilingual-mpnet-base-v2)  $MRR = 0,950$  jämfört med syntaktisk sökning med BM25 som uppnår  $MRR = 0,767$ . Förbättringen är störst på synonymfrågor och tvärspråkliga sökningar, vilket är precis de scenarier där befintlig sökfunktionalitet i FosWiki är som svagast. Ett viktigt fynd är att det semantiska systemet inte tappar prestanda på frågor med exakt terminologimatchning, vilket innebär att det skulle kunna fungera som ett alternativ till den nuvarande lösningen utan observerad prestationsförlust i det utvärderade testsetet.

Sammantaget visar arbetet att det är tekniskt genomförbart att automatisera identifiering av GDPR-känslig information och förbättra informationsåtervinning i en wiki-baserad dokumentationsmiljö med hjälp av lokalt körbara AI-modeller, utan beroende av molntjänster.

## 6.2 Reflektion och rekommendationer

Ett oväntat resultat var att hybridmetoden (RRF, MRR = 0,846) inte överträffade det semantiska systemet ensamt. Detta pekar på en viktig lärdom: en kombination av metoder är inte per automatik bättre än de enskilda delarna. När den ena underliggande metoden är markant svagare på en frågekategori riskerar fusionen att dra ned helhetsresultatet snarare än att lyfta det. En viktad fusion hade sannolikt gett ett bättre utfall, och detta identifieras som ett område för fortsatt arbete.

De metoder som utvärderats, Mistral 7B för GDPR-detektion och paraphrase-multilingual-mpnet-base-v2 för semantisk sökning, rekommenderas för implementation i Axians FosWiki-miljö via Ollama och FAISS, vilket uppfyller kraven på on-prem-drift och dataskydd. LLM-metoden bör presenteras som ett beslutsstöd snarare än ett automatiskt filter: ett Python-skript anropar modellen med jämna mellanrum, genererar en rapport över potentiellt känslig information och låter användaren avgöra vilka fynd som kräver åtgärd. Denna hanterbarhet är en styrka, men det är viktigt att kommunicera tydligt till användarna att systemet kan generera falska larm, exempelvis för systemkonton och interna hostnamn, för att undvika att förtroendet för verktyget urholkas.

Som ett första steg rekommenderas ett pilotscenario mot Axians faktiska FosWiki-miljö med verklig dokumentation, för att verifiera att resultaten från den syntetiska testmiljön håller i produktion.

## 6.3 Fortsatt arbete

Ett naturligt nästa steg är att koppla det semantiska söksystemet till en genererande LLM i en fullständig RAG-pipeline, där systemet inte bara returnerar relevanta dokument utan även genererar direkta svar på dokumentationsrelaterade frågor. Detta skulle öka nyttan för slutanvändare avsevärt och är den logiska fortsättningen på det retrieval-steg som utvärderats i detta arbete.

För hybridmetoden bör en viktad fusion undersökas, där det semantiska systemet ges högre vikt på frågor utan exakta ordmatchningar. Testdatat för GDPR-detektion bör utökas med fler dokument och entitetstyper för att möjliggöra statistiskt generaliserbara slutsatser, och en utvärderingsmetod med partiell strängmatchning bör övervägas för att ge en rättvisare bild av LLM-metodens faktiska prestanda.

Slutligen, med tanke på den snabba utvecklingen inom både AI och regulatoriska krav, är det viktigt att den implementerade lösningen är flexibel och kan anpassas i takt med att nya modeller och säkerhetsstandarder tillkommer. Vidare forskning rekommenderas för att undersöka hur finjusterade modeller eller domänspecifika embeddings kan stärka systemets precision ytterligare i en produktionsmiljö.

## Källförteckning

- [1] Kiniti S, Standing C. Wikis as knowledge management systems: issues and challenges. *Journal of Systems and Information Technology*. 2013;15(2):189–201. Tillgänglig vid: [https://www.researchgate.net/publication/263168150\\_Wikis\\_as\\_knowledge\\_management\\_systems\\_Issues\\_and\\_challenges](https://www.researchgate.net/publication/263168150_Wikis_as_knowledge_management_systems_Issues_and_challenges)
- [2] Foswiki. About Foswiki [Internet]. 2025 [citerad 2026-03-24]. Tillgänglig vid: <https://foswiki.org/Home/About>
- [3] Foswiki. Topic Markup Language [Internet]. [citerad 2026-03-24]. Tillgänglig vid: <https://foswiki.org/System/TopicMarkupLanguage>
- [4] Europeiska unionen. Allmänna dataskyddsförordningen (GDPR) [Internet]. 2022 [citerad 2026-03-24]. Tillgänglig vid: <https://eur-lex.europa.eu/SV/legal-content/summary/general-data-protection-regulation-gdpr.html>
- [5] Lison P, Pilán I, Sanchez D, Batet M, Øvrelid L. Anonymisation models for text data: state of the art, challenges and future directions. I: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. 2021. s. 4188–4203. Tillgänglig vid: <https://aclanthology.org/2021.acl-long.323>
- [6] Microsoft. Presidio: open source framework for detecting and anonymizing sensitive information [Internet]. GitHub; 2021 [citerad 2026-03-25]. Tillgänglig vid: <https://github.com/microsoft/presidio>
- [7] Microsoft. Presidio: spaCy/Stanza NLP engine [Internet]. 2024 [citerad 2026-04-30]. Tillgänglig vid: [https://microsoft.github.io/presidio/analyzer/nlp\\_engines/spacy\\_stanza/](https://microsoft.github.io/presidio/analyzer/nlp_engines/spacy_stanza/)
- [8] Zaratiana U, Tomeh N, Holat P, Charnois T. GLiNER: Generalist model for named entity recognition using bidirectional transformer. I: *Proceedings of NAACL 2024*. 2024. Tillgänglig vid: <https://aclanthology.org/2024.naacl-long.300>
- [9] Honnibal M, Montani I. spaCy: Industrial-strength natural language processing [Internet]. Explosion AI; 2017 [citerad 2026-04-30]. Tillgänglig vid: <https://spacy.io>
- [10] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using siamese BERT-networks [Internet]. arXiv; 2019 [citerad 2026-03-25]. Tillgänglig vid: <https://arxiv.org/abs/1908.10084>
- [11] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks [Internet]. arXiv; 2021 [citerad 2026-03-25]. Tillgänglig vid: <https://arxiv.org/abs/2005.11401>

- [12] Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, et al. Retrieval-augmented generation for large language models: a survey [Internet]. arXiv; 2023 [citerad 2026-03-15]. Tillgänglig vid: <https://arxiv.org/abs/2312.10997>
- [13] Jégou H, Douze M, Johnson J. Faiss: a library for efficient similarity search [Internet]. Facebook Engineering; 2017 [citerad 2026-03-24]. Tillgänglig vid: <https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/>
- [14] Chroma. Open-source data infrastructure for AI [Internet]. 2024 [citerad 2026-03-19]. Tillgänglig vid: <https://www.trychroma.com/>
- [15] Robertson S. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*. 2010;3(4):333–389. Tillgänglig vid: [https://www.researchgate.net/publication/220613776\\_The\\_Probabilistic\\_Relevance\\_Framework\\_BM25\\_and\\_Beyond](https://www.researchgate.net/publication/220613776_The_Probabilistic_Relevance_Framework_BM25_and_Beyond)
- [16] Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, Letman A, et al. The Llama 3 herd of models [Internet]. arXiv; 2024 [citerad 2026-03-24]. Tillgänglig vid: <https://arxiv.org/abs/2407.21783>
- [17] Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas D de las, et al. Mistral 7B [Internet]. arXiv; 2023 [citerad 2026-03-24]. Tillgänglig vid: <https://arxiv.org/abs/2310.06825>
- [18] Ollama. Ollama [Internet]. GitHub; 2024 [citerad 2026-03-25]. Tillgänglig vid: <https://github.com/ollama/ollama>
- [19] Friebely A. Analyzing the efficacy of Microsoft Presidio in identifying social security numbers in unstructured text [masteruppsats]. Utica University; 2022. Tillgänglig vid: <https://www.proquest.com/openview/31254af0453136664db6291485242df8>
- [20] Cormack GV, Clarke CLA, Buettcher S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. I: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2009. s. 758–759. Tillgänglig vid: <https://doi.org/10.1145/1571941.1572114>
- [21] Voorhees EM. The TREC-8 question answering track report. I: Voorhees EM, Harman DK, redaktörer. *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*. 1999. Tillgänglig vid: [https://www.researchgate.net/publication/2929514\\_The\\_TREC-8\\_question\\_answering\\_track\\_report](https://www.researchgate.net/publication/2929514_The_TREC-8_question_answering_track_report)

- [22] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*. 2009;45(4):427–437. Tillgänglig vid: [https://www.researchgate.net/publication/222674734\\_A\\_systematic\\_analysis\\_of\\_performance\\_measures\\_for\\_classification\\_tasks](https://www.researchgate.net/publication/222674734_A_systematic_analysis_of_performance_measures_for_classification_tasks)



## Bilagor

### Bilaga A – Mätvärden för GDPR-detektion

Tabell A.1: Aggregerade mätvärden för de tre detektionsmetoderna.

Metod	Precision	Recall	F1-score
Regex	0,889	0,533	0,667
Presidio (sv)	0,895	0,567	0,694
LLM (Mistral 7B)	0,833	1,000	0,909

Tabell A.2: Antal hittade entiteter och recall per entitetstyp.

Entitetstyp	Totalt	Regex hittade	Presidio hittade	Mistral hittade	Recall Regex	Recall Presidio	Recall Mistral
IP-adress	8	8	8	8	1,000	1,000	1,000
Lösenord	8	0	0	8	0,000	0,000	1,000
E-postadress	5	4	5	5	0,800	1,000	1,000
Personnamn	4	0	2	4	0,000	0,500	1,000
API-nyckel	2	0	0	2	0,000	0,000	1,000
Personnummer	2	2	2	2	1,000	1,000	1,000
Telefonnummer	2	1	1	2	0,500	0,500	1,000

Tabell A.3: Regex:s hämtningsresultat för 12 dokument, med antal korrekta svar enligt facit, totalt antal hittade, sanna positiva (TP), falska positiva (FP) samt falska negativa (FN) per dokument.

### Regex

Dok	Facit	Hittade	TP	FP	FN
1	4	2	2	0	2
2	3	1	1	0	2
3	3	3	2	1	1
4	2	1	1	0	1
5	0	0	0	0	0
6	3	2	2	0	1
7	4	2	2	0	2
8	3	1	1	0	2
9	3	3	2	1	1
10	2	1	1	0	1
11	0	0	0	0	0
12	3	2	2	0	1

Tabell A.4: Presidios hämtningsresultat för 12 dokument, med antal korrekta svar enligt facit, totalt antal hittade, sanna positiva (TP), falska positiva (FP) samt falska negativa (FN) per dokument.

**Presidio (sv+en)**

Dok	Facit	Hittade	TP	FP	FN
1	4	2	2	0	2
2	3	1	1	0	2
3	3	1	1	0	2
4	2	2	1	1	1
5	0	0	0	0	0
6	3	3	2	1	1
7	4	3	3	0	1
8	3	1	1	0	2
9	3	2	2	0	1
10	2	2	2	0	0
11	0	0	0	0	0
12	3	2	2	0	1

Tabell A.5: Mistral 7B:s hämtningsresultat för 12 dokument, med antal korrekta svar enligt facit, totalt antal hittade, sanna positiva (TP), falska positiva (FP) samt falska negativa (FN) per dokument.

**LLM (Mistral 7B)**

Dok	Facit	Hittade	TP	FP	FN
1	4	5	4	1	0
2	3	3	3	0	0
3	3	3	3	0	0
4	2	3	2	1	0
5	0	0	0	0	0
6	3	4	3	1	0
7	4	5	4	1	0
8	3	3	3	0	0
9	3	3	3	0	0
10	2	4	2	2	0
11	0	0	0	0	0
12	3	3	3	0	0

Tabell A.6: Samtliga falska positiva (FP) och falska negativa (FN) per metod.

Metod	Typ	Värde	Notering
Regex	FP	2024-03-15	Datumformat matchat som personnummer (dok 3)
Regex	FP	2024-05-20	Datumformat matchat som personnummer (dok 9)
Regex	FN	Axians2024!, backup_secret_99, SuperSecret123, Winter2024!, log@Service99, corp@VPN2024, Prod\$secure!	Lösenord utan fast format
Regex	FN	Erik Svensson, Anna Lindqvist, James Fletcher, Sarah Donnelly	Personnamn kräver NER
Regex	FN	sk-prod-8f3a2b1c9d4e, sk-live-4c7e9f2a1b3d	API-nycklar missas
Regex	FN	+44 7700 900456	Telefonnummer
Regex	FN	vpn@secure2024	E-post utan standard-TLD
Presidio (sv+en)	FP	systemagaren	Gemensamt substantiv flaggat som PERSON (dok 4)
Presidio (sv+en)	FN	Axians2024!, backup_secret_99, SuperSecret123, Winter2024!, log@Service99, corp@VPN2024, Prod\$secure!	Lösenord utan fast format
Presidio (sv+en)	FN	070-123 45 67	Telefonnummer ej i spaCy-modell
Presidio (sv+en)	FN	vpn@secure2024	E-post utan standard-TLD

Presidio (sv+en)	<b>FN</b>	sk-prod-8f3a2b1c9d4e, sk-live-4c7e9f2a1b3d	API-nycklar missas
LLM (Mistral)	<b>FP</b>	admin	Systemkonto flaggat som personnamn (dok 1)
LLM (Mistral)	<b>FP</b>	vpn.axians.internal	Internt hostname flaggat som e-post (dok 4)
LLM (Mistral)	<b>FP</b>	dbadmin	Systemkonto flaggat som personnamn (dok 6)
LLM (Mistral)	<b>FP</b>	sysadmin	Systemkonto flaggat som personnamn (dok 7)
LLM (Mistral)	<b>FP</b>	/home/deploy/.ssh/id_rsa, vpn.techcorp.internal	Sökväg/hostname flaggade som känsliga identifierare (dok 10)

## Bilaga B – Mätvärden för söksystem

Tabell B.1: Aggregerade mätvärden för de tre söksystemen. Fetstil markerar bästa värde per kolumn.

System	MRR	P@1	R@1	P@3	R@3	P@5	R@5	R@10
BM25 (syntaktisk)	0,767	0,700	0,225	0,433	0,413	0,280	0,442	0,500
<b>Semantisk (mpnet)</b>	<b>0,950</b>	<b>0,900</b>	<b>0,288</b>	<b>0,750</b>	<b>0,713</b>	<b>0,530</b>	<b>0,838</b>	<b>0,875</b>
Hybrid (RRF)	0,846	0,750	0,242	0,500	0,467	0,420	0,650	0,863



TRITA-CBH-GRU-2026:131

[www.kth.se](http://www.kth.se)