UPPSALA
UNIVERSITET

UPTEC-STS-26037

Examensarbete 30 hp

Juni 2026

Hybrid-AI för finansiell informationsutvinning

En RAG-baserad metod med evidensbaserat beslutsfattande

Marcus Arpe
Vanja Natvig

Civilingenjörsprogrammet i system i teknik och samhälle

UPPSALA
UNIVERSITET

Hybrid AI for Financial Information Extraction
En RAG-baserad metod med evidensbaserat beslutsfattande

Marcus Arpe
Vanja Natvig

Abstract

In this study, the possibility of automatically identifying, extracting, and validating financial Key Performance Indicators (KPIs) in Swedish annual reports was examined. Annual reports often contain information that is semi-structured, with tables, running text, notes, etc., which makes information extraction both time-consuming and difficult to standardise. The study combines document understanding, adaptive chunking, semantic- and lexical retrieval, and extraction methods to locate and extract KPIs from financial documents.

Before examining any retrieval methods, Docling was used to convert the semi-structured PDF documents into a structured representation. Thereafter, the content was segmented into text-, table- and group-chunks and converted into vector representations by using an embedding model, BGE-M3, which are then stored in the vector database Milvus, for hybrid-based retrieval. For the table extraction, a rule-based method was used, and for the text- and group-chunks, a fine-tuned XLM-RoBERTa model was used. To verify and validate the extracted values, a triangulation method was used, whereby multiple independent sources of evidence within the same annual report are compared. The system was trained on a goldset built from 29 annual reports, covering 62 different KPIs, and then tested on a goldset containing KPIs from 15 unseen annual reports. The retrieval was evaluated by the evaluation measures recall, Mean Reciprocal Rank (MRR), primary first, primary@20 and precision@20.

For the best configuration, querybased text with rerank, the test results with an MRR-score of 0.873 and a recall@20-value of 0.970 for the top 20 candidates, indicate that it was possible to find relevant chunks with the examined retrieval method. The test results of the table-extraction method showed an Exact Match (EM) in 97% of the cases, and for the text-extraction method, an EM in 66% of the cases was achieved. Both have matching F1 scores. Depending on the structure of the table, different results were obtained. For the best configuration, tables with only years in its columnheads showed a result EM in 99% of the cases, while columnheads without years showed EM in 81%. The triangulation method resulted in an EM score of 70% for the testset with the querybased text with rerank configuration. All the configurations resulted in a majority of Clear winners, where the triangulation results shows a large margin between the highest and second highest value.

In summery, the study shows that hybrid retrieval methods in combination with evidencebased triangulation can improve the reliability of information extraction from financial documents, with potential applications within credit risk and anomaly detection within the bank.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala

Handledare: Richard Henricsson Ämnesgranskare: Gustav Eriksson

Examinator: Elísabet Andrésdóttir

Populärvetenskaplig sammanfattning

En årsredovisning innehåller information om ett företags ekonomiska ställning för ett givet räkenskapsår. Att analysera årsredovisningar innebär bland annat att undersöka specifika nyckeltal (KPI:er), vilket idag i stor utsträckning sker manuellt. Eftersom årsredovisningar ofta varierar i struktur och innehåll är det manuella arbetet tidskrävande och svårt att standardisera. Utvecklingen av stora språkmodeller (LLM:s) har möjliggjort automatiserad informationsutvinning. Dessa modeller är särskilt robusta för dokument som innehåller en tydlig genomgående struktur.

En problematik med dokument som årsredovisningar är den varierande kontextrepresentationen, som tabeller, löpande text och bilder, vilket försvårar den automatiska informationsutvinningen. Detta beror på att olika extraktionsmetoder lämpar sig för olika kontextrepresentationer. Genom en kombination av extraktionsmetoder och retrievalbaserade strategier kan man först identifiera var i dokumentet relevant information finns, och därefter välja lämplig extraktionsteknik beroende på kontextrepresentation.

Denna studie har undersökt hur en RAG-baserad pipeline kan användas för att identifiera, lokalisera och extrahera finansiella KPI:er från årsredovisningar. Retrieval-Augmented Generation (RAG) är en teknik inom AI som möjliggör att språkmodeller kan söka efter specifik extern information utifrån en ställd sökfråga. Studiens pipeline delas in i flera individuella steg där respektive steg hanterar ett visst moment av KPI:ns utvinning.

För att möjliggöra en automatiserad hantering av PDF-dokument behöver det ofta genomgå en dokumentkonvertering, vilket avser processen att omvandla dokumentets format. Docling är ett verktyg för dokumentkonvertering som i denna studie användes för att konvertera ostrukturerade årsredovisningar till strukturerade hierarkiska representationer, med respektive delar som tabeller, textblock, rubriker och sidnummer.

För att kunna söka efter efterfrågad information i årsredovisningen användes en adaptiv segmentering som delar upp Doclingresultatet i mindre segment innehållande sammanhängande information. Detta underlättar efterföljande informationssökning och extraktion för tabeller respektive text.

En hybrid retrievalmodell (BGE-M3), tränad för semantisk sökning och retrieval, användes för att söka efter efterfrågat KPI. Modellen bygger på flera sökmetoder, som semantiska likheter och lexikala matchningar för att identifiera korrekt efterfrågat segment. Studien utvärderade även hur olika varianter av retrievaltexter (den text som sökfrågan jämförs med) och reranking (omarrangering av segment) påverkar systemets förmåga att identifiera relevanta segment.

För att extrahera värdet från de identifierade segmenten användes två separata extraktionsmetoder beroende på segmenttyp (textsegment, tabellsegment). Eftersom tabellsegment är strukturerat uppbyggda används en regelbaserad extraktion. Textsegment innehåller fler språkliga variationer, vilket försvårar den semantiska förståelsen. Därför användes en finjusterad QA-modell (XLM-RoBERTa-base-squad2).

Triangulering är en metod för att validera, öka trovärdigheten och reducera bias för ett värde eller beslut. Detta genomförs genom att kombinera och jämföra information från olika källor, metoder och perspektiv. Triangulering i denna studie avser en aggregering av flera oberoende evidenskällor (segment) för att öka trovärdigheten för det slutgiltiga värdet.

Resultaten visar att en hybridbaserad retrievalmodell, med en kombinerad sökning av semantisk- och lexikal likhet generellt presterar bättre än modeller som enbart använder sig av semantisk likhetssökning. Detta betyder att en kombinerad sökning lämpar sig väl för domäner som finansiella dokument. Resultaten visar även att reranking har en

påverkan på segmentens ordning, men påvisar mindre effekt på det totala antalet identifierade segment. Vidare visar resultaten att skillnader i retrievaltext har en påverkan på retrievalstegets identifieringsförmåga.

Extraktionsresultaten visar generellt god prestanda, där den regelbaserade tabellextraktionen uppvisar en högre extraktionsförmåga jämfört med textextraktionen. Ett strukturerat format möjliggör därmed en effektivare extraktion. Tabellextraktionen visar en högre extraktionsförmåga för tabeller med enbart år i kolumnrubrikerna jämfört med tabeller utan år. Detta indikerar att tabeller utan år i kolumnrubrikerna behöver en mer kontextuell förståelse över sambanden mellan rader och kolumner än de med bara år.

Trianguleringsresultaten visar en relativ hög grad av överensstämmelse med det korrekta värdet. Resultaten visar även att skillnader mellan källor förekommer, vilket visar på att finansiell information ofta uttrycks på olika sätt och motiverar implementeringen av en trianguleringsbaserad valideringsmetod.

Sammanfattningsvis visar studien att en hybridretrieval-metod i kombination med extraktion och triangulering kan öka tillförlitligheten i informationsutvinning från finansiella dokument.

Innehåll

1 Introduktion

Årsredovisningar används som underlag för att fatta viktiga beslut, exempelvis för kreditriskbedömning [1]. Genom att studera ekonomiska KPI:er kan det ge insikt i företags ekonomiska hälsa och prestationer [21]. Sådana analyser kan exempelvis innebära att jämföra KPI:er över tid för att upptäcka trender och förändringar, eller att jämföra dem med branschens medelvärden för att bedöma ett företags prestationer [31]. Årsredovisningar kan variera i både struktur och innehåll, vilket gör det manuella arbetet för att extrahera information ur dessa dokument tidskrävande och svårt att standardisera [37]. Dokumenten består ofta av flera typer av kontext, där information kan förekomma i rubriker, tabeller och löpande text, vilket ytterligare försvårar en enhetlig hantering [45].

Utvecklingen av stora språkmodeller (LLMs) och dokumentförståelsemodeller har skapat möjligheter för automatiserad extraktion av information som är både semantiskt och kontextuellt korrekt [12]. Den mest lämpade extraktionsmetoden varierar beroende på kontextens struktur. För tabellbaserad information används ofta strukturerade metoder, exempelvis radklassificering [8], medan för extraktion från löptext kan LLM-baserade fråge-svar-modeller (QA) vara mer effektiva [44].

Genom att kombinera dessa extraktionsmetoder med retrieval-baserade strategier kan man först identifiera var i dokumentet relevant information finns [33], och därefter välja lämplig extraktionsteknik beroende på om informationen återfinns i tabeller eller löptext. Detta möjliggör en mer systematisk, effektiv och tillförlitlig hantering av finansiell information från årsredovisningar.

Finansiella dokument, som exempelvis årsredovisningar, består ofta av en kombination av olika strukturformat, som text, tabeller, bilder, diagram etc. KPI:er och annan viktig information kan därför presenteras i många olika format, vilket försvårar användningen av traditionella metoder för informationsutvinning som ofta är anpassade för en viss typ av dokumentstruktur [4].

Finansiell information är i många fall kontextberoende, vilket innebär att närliggande information kan vara nödvändig för att förstå innebörden av ett specifikt KPI [7]. KPI:er kan förekomma i varierande strukturella och semantiska kontexter såsom rubriker, hierarkiska nivåer och tabellrader. Detta gör att en korrekt extraktion inte enbart förutsätter korrekt identifiering av KPI:t i sig, utan även att den omkringliggande informationen har stor betydelse för korrekt utvinning.

För att hantera detta har metoder som tar hänsyn till semantisk betydelse vid informationsutvinning och analys av finansiella dokument blivit allt vanligare [38]. Metoderna använder vektorrepresentationer av den finansiella texten för att identifiera semantiska likheter. På så sätt kan relevanta segment hittas i dokumentet innan extraktionssteget sker [20].

Eftersom hög säkerhet och korrekthet är särskilt viktigt inom banksektorn kräver införandet av maskininlärningsmodeller en noggrann utvärdering av modellens robusthet och tillförlitlighet. I detta sammanhang blir triangulering ett centralt begrepp. Trianguleringen bidrar även till att stärka förutsättningarna för framtida tillämpningar inom anomalidetektion i finansiella dokument, genom att möjliggöra identifiering av avvikelser och inkonsekvenser mellan olika källor.

Ett tidigare arbete inom detta område undersökte automatisk extraktion av finansiella KPI:er från årsredovisningar med hjälp av olika maskininlärningsmetoder. Resultaten visade att en finjusterad extraktiv modell kunde uppnå cirka 95 % Exact Match när rätt kontext redan var identifierad. Vid extraktion från hela årsredovisningar sjönk resultatet till 53-60 % Exakt Match [51].

Resultaten tyder på att den huvudsakliga utmaningen inte ligger i själva extraktionssteget, utan i att lokalisera rätt information i dokumentet. Denna studie bygger därför vidare på det tidigare arbetet genom att undersöka hur retrieval-metoder kan användas för att identifiera relevanta dokumentsegment innan extraktion genomförs.

2 Syfte

Syftet med denna studie var att undersöka hur väl en RAG-baserad pipeline kan identifiera, lokalisera och extrahera finansiella KPI:er från årsredovisningar. Arbetet omfattar utveckling av retrieval-, extraktions- och trianguleringsmetoder. Vidare undersöks hur informationens placering och struktur i dokumentet påverkar möjligheten att korrekt identifiera och extrahera KPI:er.

2.1 Forskningsfrågor

1) Hur kan en RAG-liknande pipeline användas för att extrahera och validera KPI-värden från finansiella rapporter på ett robust sätt?

2) Hur påverkar olika retrieval-konfigurationer kvaliteten på kandidatsegment för KPI-extraktion?

3) Hur skiljer sig extraktion mellan tabellbaserade och textbaserade segment?

4) Hur kan flera kandidatvärden från olika källor aggregeras för att välja ett robust slutvärde?

3 Teori

3.1 Informationsutvinning

Informationsutvinning avser processen att automatiskt omvandla ostrukturerad eller semistrukturerad text till strukturerad information [49].

I denna studie syftar informationsutvinning på identifiering och extrahering av KPI:er från text och tabeller ur årsredovisningar.

3.2 Dokumentkonvertering

Dokumentkonvertering avser processen att omvandla formatet på ett dokument, som att konvertera från en ostrukturerad representation till en strukturerad representation. Detta återskapar dokumentets logiska struktur och underlättar automatisk bearbetning av dokument [63].

Vid dokumentkonvertering analyseras ett dokument genom att dess visuella och strukturella komponenter identifieras såsom textblock, rubriker, tabeller, bilder och figurer. Därefter sker en layoutanalys där visuella egenskaper som fontstorlek, fetstilta ord, placering på sidan och numrerade avsnitt analyseras. Informationen används för att klassificera dokumentets innehåll och identifiera varje beståndsdels funktion. Slutligen organiseras de identifierade delarna till en strukturerad representation i en hierarkisk struktur [56].

Docling är ett ramverk för dokumentkonvertering som omvandlar ett dokument till en strukturerad hierarkisk representation med respektive delar som tabeller, textblock, rubriker och sidnummer [19].

Ett typiskt dokument som docling kan appliceras på är PDF-dokument, vilka utgörs av textpositioner, koordinater, objekt, fonter och layoutinformation, snarare än en dokumentstruktur som kan bearbetas av automatiserade system. Detta innebär att den semantiska och logiska strukturen i dokumentet inte är explicit representerad och därför inte direkt tillgänglig för maskinell tolkning [62].

Docling är särskilt relevant för denna studie då hela pipelinen bygger på den struktur som skapats vid dokumentkonverteringen. Detta innebär att verktyget är avgörande för kvaliteten i pipelinens efterföljande steg.

3.3 Adaptiv segmentering

Segmentering handlar om att dela upp ett dokument i mindre delar, så kallade segment. Adaptiv segmentering är en typ av segmentering som anpassas efter dokumentets struktur, innehåll, semantik och layout. På så vis kan den semantiska kontexten och den sammanhängande informationen behållas i varje segment.

Syftet med en adaptiv segmentering är att bevara den semantiska kontexten och sammanhängande informationen inom varje segment, vilket är särskilt viktigt vid bearbetning av komplexa dokument som årsredovisningar [16].

I denna studie är adaptiv segmentering centralt, då dokumentets innehåll delas in i mindre kontextspecifika segment, vilken underlättar informationsutvinningen i pipelinens senare steg.

3.4 Tokens och tokenisering

Tokenisering är en process där text delas upp i mindre delar kallade tokens. En token kan se olika ut beroende på tokeniseringsmodell och kan bestå av enskilda tecken, delar av ord eller hela ord. Dessa tokens konverteras därefter till numeriska representationer som språkmodeller kan bearbeta.

Denna studie använder sig av subword-baserad tokenisering, vilket innebär att ord kan delas upp i mindre delar. Detta möjliggör en effektiv hantering av språklig variation och okända ord, vilket är särskilt relevant för finansiella texter där terminologi kan variera [27].

Tokenisering är särskilt relevant i denna studie då både retrieval- och extraktionsstegen bygger på transformbaserade modeller. En effektiv tokenisering leder till att domänspecifika termer kan representeras på ett konsekvent sätt, vilket underlättar både semantisk sökning och korrekt informationsutvinning.

3.5 Embeddings

Embedding är en metod för att representera text som högdimensionella numeriska vektorer, där textens semantik fångas upp och representeras i form av en vektor. Denna metod möjliggör en semantisk informationssökning mellan texter genom att jämföra avstånd mellan vektorer snarare än exakta matchningar. Genom denna representation placeras semantiskt liknande texter nära varandra i vektorrummet. Embeddings används därför som en central komponent i moderna informationssökningssystem, där relevanta segment kan identifieras även vid språklig variation för samma koncept [6].

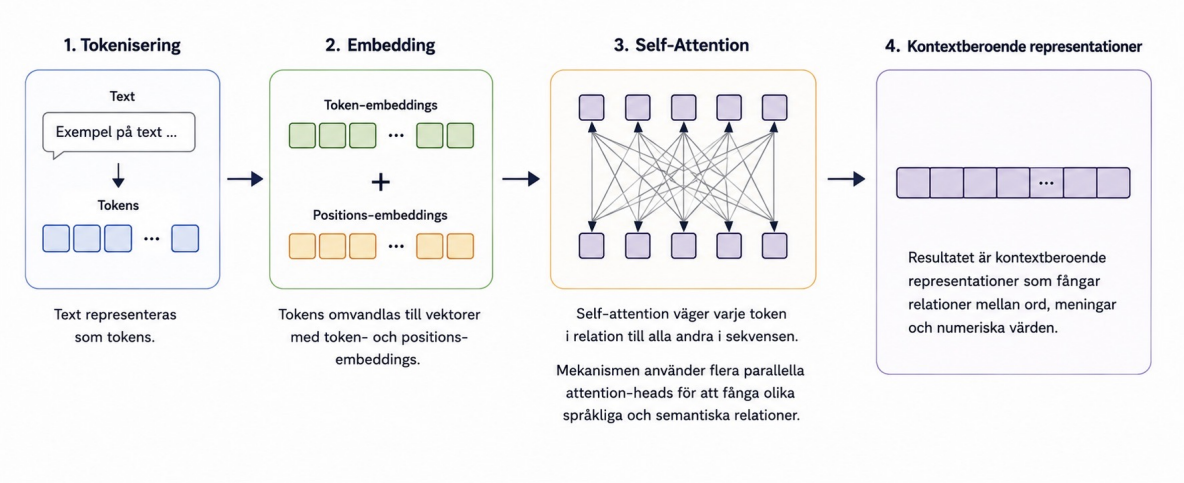
I denna studie är detta särskilt relevant eftersom retrievalsteget använder embeddings för att identifiera relevanta segment innan extraktionssteget utförs.

3.6 Transformermodeller

En transformermodell är ett neuralt nätverk som analyserar relationer mellan delar av sekventiell data, exempelvis text. Detta sker genom att skapa kontextberoende representationer av innehållet genom self-attention [53]. Detta visualiseras i Figur 1.

I denna arkitektur representeras texten i tokens. Dessa omvandlas till vektorer genom token-embeddings och positions-embeddings, vilket möjliggör att modellen kan bevara ordningsföljden samtidigt som den bearbetar sekvensen parallellt.

Self-attention är en mekanism som väger varje token i relation till övriga tokens i sekvensen, vilket gör att modellen kan fokusera på kontextens relevanta delar. Mekanismen delas upp i flera parallella attention-heads som används för att fånga olika delar av språkliga och semantiska relationer, vilket förbättrar modellens representationsförmåga [53].



Figur 1: *Transformermodellens arktitektur. Bild genererad med ChatGPT (OpenAI, 2026).*

Denna arkitektur utgör grunden för moderna embeddingmodeller eller språkmodeller som BGE-M3, Multilingual-e5-large och XLM-RoBERTa [15], [42], [57].

3.6.1 BGE-M3

BGE-M3 är en flerspråkig embeddingmodell som används för semantisk sökning och retrieval. Modellen tränas på data som innehåller en sökfråga, relevanta (positiva) och irrelevanta (negativa) segment, vilket lär modellen att placera relevanta segment nära sökfrågan och irrelevanta segment längre bort i vektorrummet. Detta gör modellen väl lämpad för retrievaluppgifter. BGE-M3 stödjer både dense retrieval (semantiska likheter via embeddings) och sparse retrieval (lexikal matchning), vilka kan kombineras till en hybrid retrieval-metod. Detta gör denna modell särskilt effektivt för domäner som finansiella dokument, med KPI:er, siffror och återkommande termer [10].

Denna modell är relevant för studien eftersom den används för att skapa embeddings av sökfråga och segment, vilket möjliggör den hybrida retrievaln.

3.6.2 Multilingual-e5-Large (E5)

Multilingual-e5-Large är, likt BGE-M3, en flerspråkig transformerbaserad embeddingmodell, utvecklad främst för semantisk sökning och retrieval. Modellen är tränad på samma sätt som BGE-M3, vilket gör den väl lämpad för retrievaluppgifter. Till skillnad från tidigare modell är E5 optimerad för dense retrieval [57].

E5 används som en jämförelsemodell för att undersöka hur valet av embeddingmodell påverkar retrievalprestandan.

3.6.3 BERT, RoBERTa och XLM-RoBERTa

BERT (bidirectional Encoder Representations from Transformers) är en transformbaserad språkmodell som lär sig kontextuella representationer genom maskerad språkmodellering. Detta innebär att vissa ord i träningstexten döljs för att modellen sedan lär sig kontextens semantiska betydelse genom att förutsäga de dolda orden utifrån resterande kontext [17].

RoBERTa är en vidareutbyggnad på BERT genom förbättrad träning, dynamisk maskning (mer variationer i vilka ord som maskeras) och större datamängder ger en bättre språkförståelse och bredare generaliserbarhet [32].

XLM-RoBERTa är en variant av RoBERTa som är tränad på stora mängder data och designad för att klara av många olika språk, vilket gör den användbar för just svenska årsredovisningar [42].

Denna modell är relevant för studien då en finjustering av modellen används för att extrahera KPI-värden från löpande text.

3.7 Milvus

Milvus är en open-source vektordatabas skapad för att lagra, indexera och söka med vektorrepresentationer [36]. Milvus söker i sin databas genom att ta emot två parametrar: sökfråga, representerad som en embeddingvektor och antalet träffar (k) som ska returneras. För att kunna hantera stora mängder högdimensionella vektorer använder Milvus Approximate Nearest Neighbor-sökning (ANN), vilken är utformad för att hitta nära kandidater i vektorrummet utan att behöva beräkna exakta avstånd för alla vektorer.

Från kandidatlistan beräknas likheten mellan frågevektorn och kandidaterna med valt likhetsmått [55].

I denna studie används inre produkt eftersom den är väl lämpad för embeddingbaserade retrievalsystem.

Den inre produkten för två vektorer x och y med k dimensioner beräknas enligt Ekvation 1.

$$x \times y = \sum_{i=1}^{k} x_i y_i \tag{1}$$

Ett högre värde indikerar större likhet mellan vektorerna och därmed högre semantisk likhet mellan kandidaterna [55].

Milvus är relevant för denna studie då den används för att indexera och lagra segment så att relevanta segment snabbt kan identifieras när en sökfråga ställs.

3.8 Retrieval-Augmented System

Retrieval-Augmented System kombinerar språkmodellers generativa och extraktionsbaserade förmåga med extern informationshämtning. Istället för att enbart förlita sig på modellens interna kunskap inhämtas relevant information från en extern källa, vilket gör att svar baseras på aktuella och domänspecifika dokument. Systemet gör detta i två steg, först en sökning efter relevanta segment utifrån en specifik sökfråga, sedan används informationen för att antingen generera eller extrahera ett svar [26].

I denna studie är detta centralt då retrieval används för att lokalisera relevanta segment som efterföljande extraktionssteg använder sig av för att extrahera relevant information från. Denna RAG-baserade arkitekturen fungerar därmed som den grundläggande arkitekturen för informationsutvinningen.

3.8.1 Information Retrieval

Information Retrieval (IR) handlar om att, utifrån en given sökfråga eller informationsbehov, hämta relevanta segment. De hämtade segmenten rangordnas utifrån relevans, vilket innebär hur väl segmentet motsvarar den information som efterfrågas [24].

Traditionella IR-metoder bygger ofta på lexikal matchning. Dessa metoder är ofta termbaserade, vilket innebär att förekomsten av varje ord räknas. Dessa frekvenser används sedan för att avgöra hur relevant ett segment är i förhållande till en specifik sökfråga [23]. Dessa metoder är effektiva för dokument där terminologin i både sökfråga och dokument är densamma, men när variationer i formuleringar och synonymer förekommer blir det svårare för systemet att identifiera likheter. Detta har lett till utvecklingen av mer avancerade IR-metoder baserade på embeddings [3], [33].

I denna studie handlar retrieval om att, utifrån sökfråga, identifiera relevanta segment från årsredovisningar.

3.8.2 Lexikal sökning

Lexikal sökning (sparse retrieval) är en metod där segment matchas baserat på överlapp i exakta termer mellan en sökfråga och segmentinnehåll. Sökningen baseras på innehållets matchade ord och rangordnas baserat på vikten av termerna och hur frekvent orden förekommer. För en fråga som ”Vad är rörelseresultat år 2024?” kan ett segment som ”Rörelseresultat för 2024 uppgick till 320 MSEK” rankas högt då det innehåller flera av frågans centrala termer. Ett segment som ”EBIT för förra året var 320 MSEK” kan däremot rankas lägre trots att det innehåller liknande information, eftersom ordvalet skiljer sig från sökfrågan.

Metoden är därför effektiv när samma terminologi används i både sökfråga och segment, men begränsad när språkliga variationer eller synonymer förekommer [60].

Lexikal sökning är relevant i denna studie för att komplettera den semantiska sökningen genom att identifiera segment som innehåller exakta matchningar av KPI-relaterade termer.

3.8.3 Semantisk sökning

Semantisk sökning (dense retrieval) innebär att identifiera innehåll med liknande betydelse genom att ta hänsyn till den semantiska innebörden i både sökfråga och segment, snarare än exakta ordmatchningar. Detta utförs genom att representera text som embeddings [28].

Detta är särskilt relevant för denna studie då den semantiska sökningen utgör grunden för retrievalsökningen där både sökfråga och segment representeras som vektorer och jämförs utifrån deras semantiska likhet.

3.8.4 Hybridsökning

Hybridsökning (hybrid retrieval) är en metod som kombinerar lexikal sökning och semantisk sökning för att utnyttja styrkorna i båda tillvägagångssätten och täcka varandras brister. Lexikal sökning är effektiv på att hitta exakta ordmatchningar, men har begränsad förmåga att förstå semantisk variation. Semantisk sökning kan istället fånga semantiska likheter, men missar ibland exakta numeriska uttryck eller specifika KPI-termer eftersom semantiska likheter prioriteras. Med en hybridsökning kan därför både lexikala och semantiska likheter vägas in, vilket möjliggör en mer robust sökning [46].

För att i denna studie dra nytta av både semantiska relationer och exakta ordmatchningar kombineras en semantisk- och lexikal sökning i en hybrid sökning.

3.9 Reciprocal rank fusion (RRF)

Reciprocal Rank Fusion (RRF) är en metod för att kombinera och omvandla flera rankade listor $rank(r)$ från olika sökningar till en gemensam rankning. Metoden bygger på idén att de kandidater som rankas högt i flera listor är de mest relevanta. RRF beräknas enligt Ekvation 2, där $rank(r)$ anger segmentets position i listan och k är en konstant.

$$\text{Score}_{\text{RRF}} = \sum_{r \in R} \frac{1}{k + rank(r)} \tag{2}$$

Genom att enbart använda dokumentets position i listan gör detta listorna jämförbara mellan de olika sökmetoderna [14].

Detta är viktigt för studien då RFF används för att kombinera kandidatlistorna från de tre olika sökmetoderna till en gemensam lista.

3.10 Reranking

Reranking är en metod för att omordna segment för att skapa en förbättrad slutlig rangordning från en eller flera retrievalmetoder genom en utvärdering av segmentets relevans mot sökfrågan [41].

Reranking används i denna studie för att förbättra retrievalkvaliteten och finjustera rangordningen av segment baserat på innehåll och metadata.

3.10.1 Fuzzy matching och heuristisk rerank

För att göra systemet mer robust mot variationer i formuleringar används fuzzy matching. Fuzzy matching är en teknik för att identifiera ord eller meningar som är liknande men inte identiska, där en likhetspoäng baseras på hur lika orden/meningarna är till varandra [40].

Heuristisk rerank är en metod inom IR där de initialt hämtade segmenten omordnas med hjälp av regelbaserade, deterministiska scoring-funktioner snarare än inlärda modeller. Funktionerna kombinerar ofta faktorer som relevans för att förbättra den slutliga kvaliteten på rankingen.

Till skillnad från inlärda metoder som learning-to-rank eller cross-encoders krävs ingen träningsdata för regelbaserade metoder och byggs istället upp på explicita antaganden om vad som utgör ett relevant segment [34].

I studien används en kombination av fuzzy matching och heuristiska regler för att identifiera KPI-likheter mellan sökfråga och dokument, vilken lägger grunden till studiens reranking.

3.10.2 Cross-encoder - BGE-reranker-large

En cross-encoder är en modell som används för att beräkna relevans mellan två texter. Modellen jämför varje token i sökfrågan mot varje token i segmentet för att avgöra relevans. Jämförelsen sker via self-attention i transformern och upprepas i flera lager. Varje token uppdateras successivt med information från alla andra tokens i både sökfrågan och segmentet, vilket gör att både lokala och globala beroenden kan fångas. Resultatet blir en gemensam representation som används för att beräkna en relevanspoäng för hela segmentet [48].

BGE-reranker-large är en cross-encoder-modell som är optimerad för rerank i informationssökning. Modellen är tränad för att prediktera relevans på stora mängder sökfråga-text-par, vilket gör att den lär sig finare semantiska skillnader än vad enbart embeddingbaserade modeller kan upptäcka [65].

Som ett alternativ till den heuristiska omrankningen utvärderades även bge-cross-encoder-modellen i studien.

3.11 Informationsextraktion

Informationsextraktion innebär att identifiera och extrahera specifik information från ett dokument [25].

Beroende på dataformatet, är olika typer av extraktionsmetoder mer lämpliga än andra. Regelbaserade extraktionsmetoder är lämpade för information som återfinns i tabeller eftersom tabeller ofta innehåller tydligt definierade strukturella relationer mellan radetiketter, kolumner och numeriska värden [52], [47].

För löpande text krävs ofta en förståelse för semantiska och kontextuella relationer mellan ord och meningar, därför är transformer-baserade språkmodeller lämpliga för extraktion av den formattypen [53].

Informationsextraktion är relevant i denna studie för att identifiera finansiella KPI:er och extrahera tillhörande värden från de segment som hämtats under retrievalsteget.

3.11.1 Regelbaserad extraktion

Regelbaserad extraktion innebär att information extraheras utifrån fördefinierade regler, mönster och logik. Reglerna kan beskriva hur information brukar se ut, var den oftast hittas och hur den ska identifieras [59].

Reguljära uttryck (regex) är en deterministisk, regelbaserad metod där reguljära uttryck används för att beskriva specifika mönster i en textsträng och på så vis identifiera exempelvis årtal, radetiketter och kandidatvärden. Istället för att söka efter en exakt textsträng, definieras regler för vilka tecken som får förekomma, hur många gånger de ska förekomma och i vilken ordning de ska följa varandra. Exempelvis kan uttrycket r'`\d{4}'` användas för att matcha årtal. Uttrycket resulterar i att alla fyrsiffriga tal ger en match. Likadant kan specifika bokstavskombinationer eller andra specifika mönster

i texten hittas. Regex är användbart när information har en relativt konsekvent struktur och följer tydliga mönster [22].

Detta är särskilt relevant för denna studie då finansiella tabeller ofta innehåller liknande struktur för årtal, radetiketter och värden. I denna studie används därför en regelbaserad extraktionsmetod för att identifiera och extrahera det värde som bäst överensstämmer med det efterfrågade KPI:t i tabeller.

3.11.2 Modellbaserad extraktion

Information som förekommer i löptext, uppvisar ofta stor variation i formuleringar och uttryck, vilket gör regelbaserade extraktionsmetoder mindre robusta.

En metod för modellbaserad extraktion är Question Answering modeller (QA-modeller). Neurala språkmodeller, som QA-modeller, är tränade för att modellera relationen mellan en fråga och ett segment. En fråga tillsammans med ett segment skickas till modellen som indata och modellen identifierar det textspann som bäst besvarar frågan.

Modellen beräknar sannolikheter för alla tokens i texten och predikterar start- och slutpositionen för det mest sannolika svarsspannet. Det svarsspann med högst sannolikhet väljs som det extraherade svaret. Detta innebär att modellen inte genererar svar i fri text, utan istället lokaliserar det mest sannolika svarsspannet direkt i det givna segmentet.

Eftersom modellen tar hänsyn till kontexten runt varje token kan relevanta svar identifieras även när varierande formuleringar förekommer. Detta beror på att modellen inte enbart matchar exakta ord, utan även lär sig semantiska samband mellan sökfrågans innehåll och kontextens olika uttryckssätt, vilket gör den robust mot språklig variation och omformuleringar [54], [64], [39].

I denna studie används QA-modellen XLM-RoBERTa för extraktion ur löpande text, för att fånga den semantiska betydelsen i språkliga varierande texter.

3.12 Utvärderingsmått (EM, F1-score, MRR)

Exact Match (EM) är ett utvärderingsmått på hur ofta det extraherade svaret faktiskt är det korrekta svaret. EM beräknas enligt Ekvation 3

$$EM = \frac{\text{Antal exakta träffar}}{\text{Totalt antal exempel}} \tag{3}$$

vilket ger ett värde på andelen prediktioner där det extraherade svaret exakt stämmer överens med facit [43].

F1-score är ett annat utvärderingsmått som kombinerar precision och recall genom att beräkna deras harmoniska medelvärde. *Precision* mäter andelen identifierade resultat som är korrekta genom att utvärdera förhållandet mellan korrekta träffar (True Positives (TP)) och det totala antalet identifierade träffar, där FP (False Positive) står för felaktiga träffar. Detta kan ses i Ekvation 4.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall är ett mått på hur stor andel av alla korrekta resultat som identifieras, se Ekvation 5,

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

där FN (False Negative) är antalet missade träffar.
F1-score beräknas sedan enligt Ekvation 6.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

Om F1-score är högt innebär det att både precision och recall är höga [11].

EM och F1-score är relevanta mått för utvärdering av både extraktion och triangulering.

Mean Reciprocal Rank (MRR) är en annan utvärderingsmetod ofta använd inom retrievalutvärdering. Den använder sig av så kallad *Reciprocal Rank* (RR), vilket är ett mått på hur tidigt i rankingen det första relevanta segmentet förekommer. RR beräknas utifrån positionen för det första relevanta resultatet i en rankad lista och beräknas enligt Ekvation 7,

$$RR = \frac{1}{rank_i} \tag{7}$$

där $rank_i$ beskriver position i.

MRR beskriver i sin tur den genomsnittliga rankingen för det första relevanta segmentet utifrån ett givet retrievalresultat. Måttet beräknas enligt följande Ekvation 8.

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{8}$$

MRR resulterar i ett värde mellan 0 och 1, där högre MRR beskriver en högre placering av det första relevanta segmentet [18], [50].

Recall, precision och MRR är relevanta mått för att utvärdera retrievalstegets prestation.

3.13 Triangulering

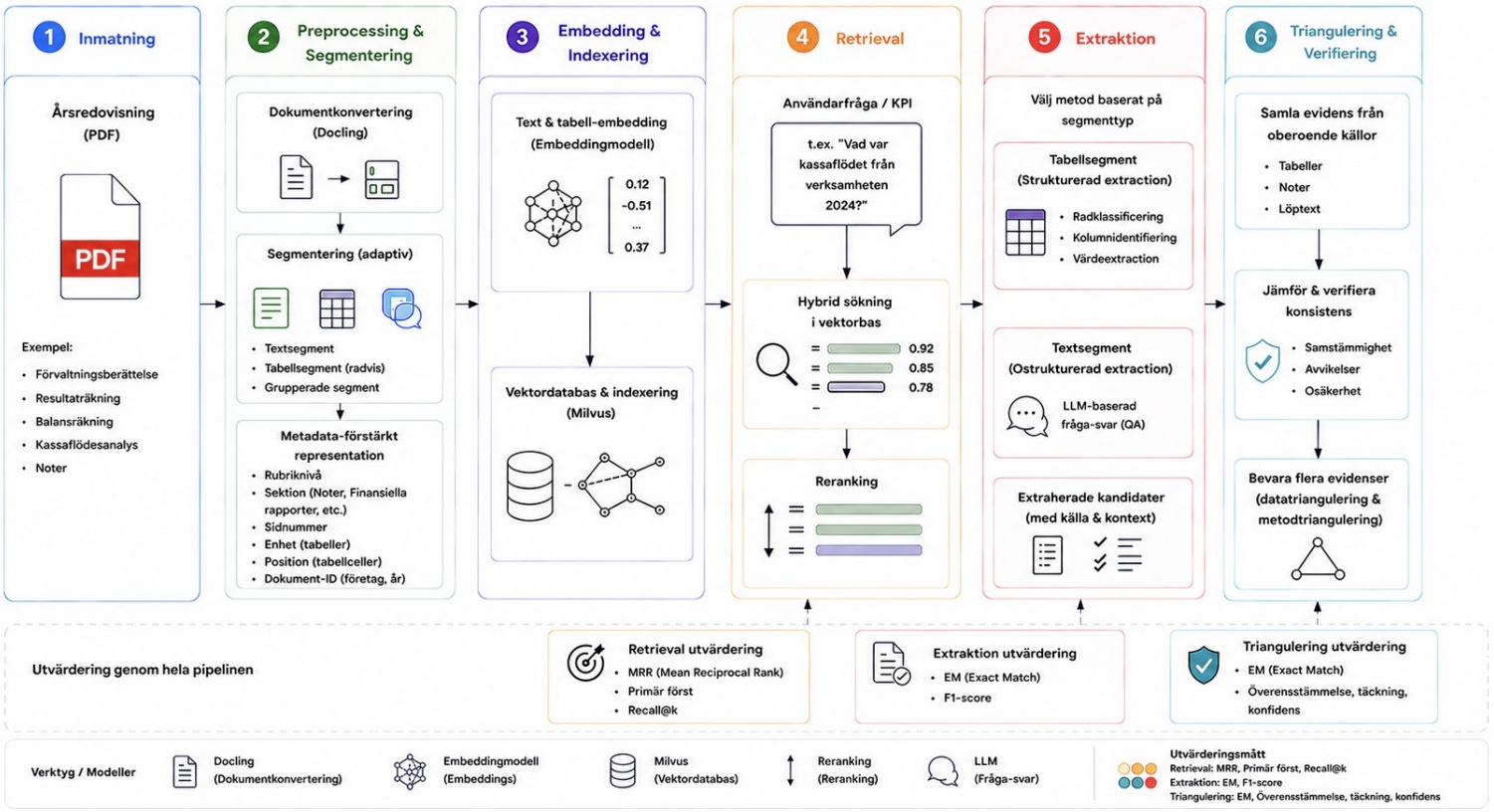
Triangulering är en metod för att validera, öka trovärdigheten och reducera bias för ett visst värde eller beslut. Detta görs genom att kombinera och jämföra information från flera olika källor, metoder eller perspektiv [58] [9]. Eftersom forskningsresultat alltid är föremål för osäkerhet och tolkning, kan tillförlitligheten stärkas genom att data hämtas och jämföras från flera oberoende evidenskällor [5].

Evidensaggregation är ett relaterat koncept som innebär att information från flera oberoende evidenskällor vägs samman för att förbättra tillförlitligheten i ett slutgiltigt beslut. Grunden i evidensaggregation är att överensstämmelse mellan flera olika källor utgör starkare evidens än information från en enskild källa, och därmed stärks tilltron för resultatet [35].

Triangulering är särskilt relevant i denna studie eftersom flera kandidatkällor kan identifieras för samma KPI. Genom att aggregera flera kandidater kan triangulering jämföra och väga samman evidens från flera källor för att utvinna det värde med starkast stöd i dokumentet.

4 Metod

Metoden för detta arbete utgår från en pipeline som visualiseras i Figur 2, där varje moment behandlas separat och bygger vidare på föregående stegs resultat. All insamlad data i studien kommer från offentligt publicerade svenska årsredovisningar.



Figur 2: *Figuren visar pipelinens sex olika steg som undersöks i studien. Den startar med en årsredovisning i PDF-format som indata och slutar med en trianguleringsverifiering av de extraherade värdena. Bild genererad med ChatGPT (OpenAI, 2026).*

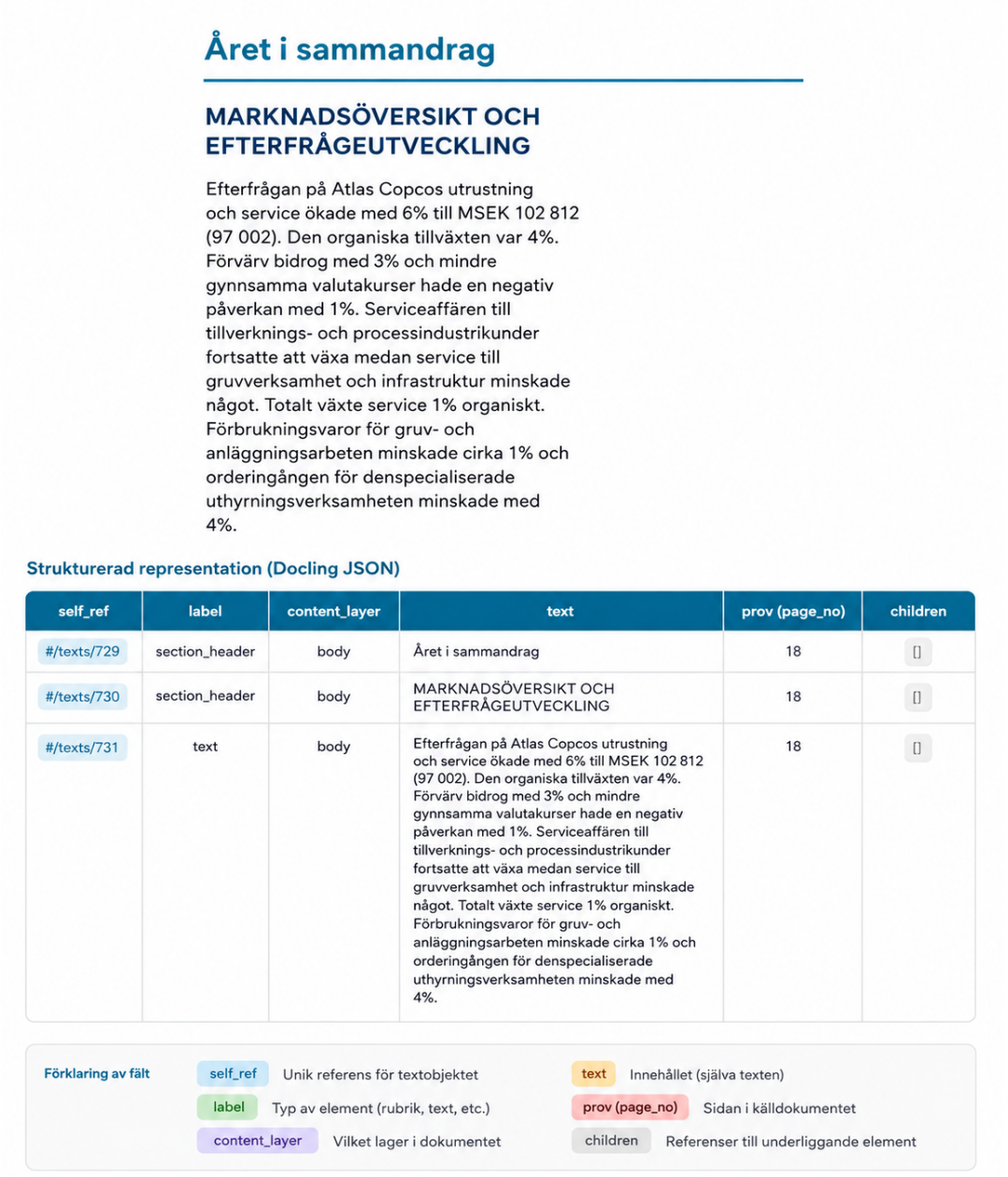
4.1 Preprocessing & Segmentering

Segmenteringen i denna studie avser processen att omvandla årsredovisningen till mindre semantiska sammanhängde enheter. Genom att göra detta möjliggörs en effektiv retrieval och efterföljande extraktion. Segmenteringen skapar därmed mer specifika segment, vilket underlättar retrievalstegets förmåga att lokalisera relevanta segment, samt extraktionsstegets förmåga att identifiera och extrahera relevant information.

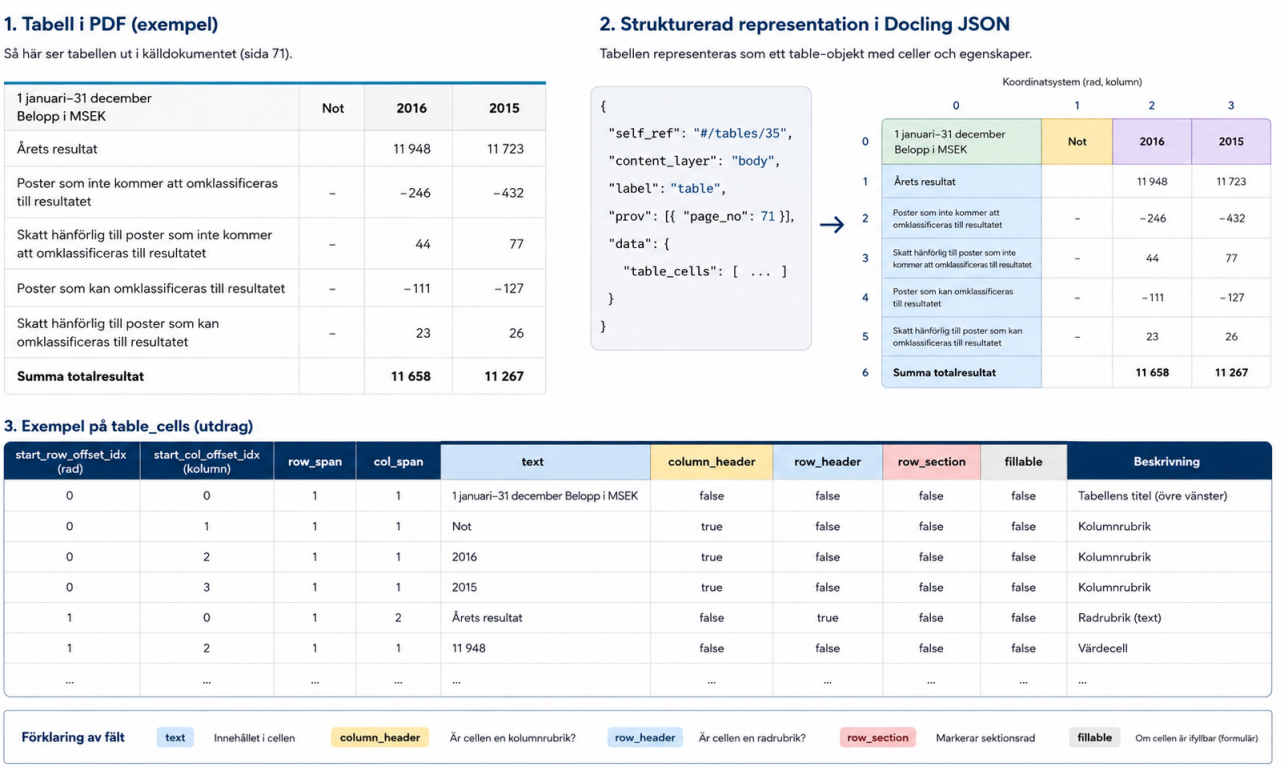
4.1.1 Docling

Docling används för att omvandla PDF-dokument till strukturerade JSON-dokument som segmenteringsmodellen kan arbeta med. Varje dokument består av en hierarkisk

struktur med de olika elementtyperna: textelement, tabellelement, bilder och figurer och grupperade element. Elementen innehåller viktig metadata som sidnummer, rubrik, dokumentnivå samt positionsinformation för tabellceller. Exempel på en pseudorepresentation för JSON-utdata och dessa respektive dela av årsredovisningen kan ses i Figur 3 och 4.



Figur 3: *Skärmavbild tagen från Atlas Copco årsredovisning 2016 med pseudorepresentation av JSON-dokument för textelement. Bild genererad med ChatGPT (OpenAI, 2026).*



Figur 4: *Skärmavbild tagen från Atlas Copco årsredovisning 2016 med pseudorepresentation av JSON-dokument för tabellelement. Bild genererad med ChatGPT (OpenAI, 2026).*

På grund av RAM-minnesbegränsningar används en segmenteringsmetod i dokumentkonvertering med Docling. Varje PDF delas in i segment av 20 sidor med hjälp av Python-biblioteket PyPDF och konverteras sedan separat oberoende av varandra. Varje skapad JSON-fil sätts därefter ihop med Doclings inbyggda sammanfogningsmetod som bevarar korsreferenser, elementordning och sidnummer över segmentgränserna. Varje sammansatt Docling-dokumentobjekt exporteras därefter till JSON-format för bearbetning i nästa del av pipelinen.

4.1.2 Adaptiv segmentering

En adaptiv segmenteringsmodell implementeras för att möjliggöra en effektiv semantisk informationssökning. Modellen i denna studie skiljer på de två segmenttyperna, text och tabeller. Anledningen till detta är att tabeller består av ett mer strukturerad format än löpande text, vilket gör att extraktionen kan anpassas för varje segmenttyp.

Årsredovisningarnas innehåll delas in i mindre segment av typerna: textsegment, tabellsegment och grupperade segment (punktlistor eller nyckel-värde-par). Varje segment innehåller viss data, varav två fält består av retrievaltext och extraktionstext. För att varje segment ska bevara årsredovisningens hierarkiska struktur sparas även dess metadata.

För att separera årsredovisningarna i relevanta segment utgår studien från följande segmenteringsprinciper.

- Ny sektionsrubrik
- Ny underrubrik
- Byte mellan text, tabell och grupp
- För tabeller sker segmentering för varje ny tabellrad. På så sätt möjliggörs en exakt hämtningsutvärdering och lokalisering på en entydig semantisk mening.
- Överstiger 300 tokens.

Om någon av dessa kriterier uppfylls påbörjas ett nytt segment. Genom detta säkerställs att varje segment är semantiskt sammanhängande.

4.1.3 Hantering av textsegment och grupperade segment

Varje textelement i Docling klassificeras som antingen sektionsrubrik, löpande text eller grupp. För textelementet ”sektionsrubrik” uppdateras segmentens kontextuella metadata, där rubriken sparas och används för att härleda överordnad sektion (exempelvis Noter, Finansiella rapporter, Förvaltningsberättelse) genom regelbaserad mappning.

För att skapa segment av den löpande texten adderas ny text successivt till det aktuella segmentet tills den maximala token-storleken uppnås på 300 tokens. Valet av 300 tokens baseras på en avvägning mellan att bevara tillräcklig kontext för retrieval och extraktion samt att begränsa mängden information i varje segment. Segmentstorleken bedöms vara tillräckligt liten för att bibehålla semantisk sammanhållning, samtidigt som den är tillräckligt stor för att innehålla relevant kontext kring de finansiella KPI-värdena.

Om ett textsegment överstiger återstående token-utrymme delas segmentet adaptivt. Den del av texten som får plats adderas till segmentet medan resterande del hanteras i efterföljande segment. När nästa segment skapas inleds det med en överlappning av de 60 tokens som föregående segment avslutades med. På så sätt minskas risken för kontextförlust. Valet av 60 tokens baseras på en avvägning mellan att bibehålla tillräcklig kontext för retrieval och extraktion och samtidigt att begränsa duplicerad information mellan segmenten.

För att undvika att textsegment delas på olämpliga platser (exempelvis mitt i en mening) implementeras en regelbaserad heuristik som, vid uppnådd tokenbegränsning, identifierar det sista förekommande skiljetecknet eller radbrytningen inom det tillåtna tokenintervallet. Om ett sådant naturligt stopp hittas delas texten där, annars sker delningen vid den maximala tokenlängden.

Grupperade segment hanteras olika beroende på om det är en lista eller nyckelvärdepar. I årsredovisningar förekommer dessa ofta i form av listade upplysningar, sammanfattande punkter, definitioner eller för att presentera data som är strukturerad i tabellformat. I denna studie hanteras dessa som textsegment. Exakta specifikationer för grupperande segment kan utläsas i Appendix A.

För varje textsegment utgör retrievaltexten hela det tillhörande textinnehållet för respektive segment. För extraktionstexten görs ett tillägg med information om företagsnamn och år vilket ger följande format:

```
DOC_ID: Företagsnamn_ÅR | TEXT: Segmenttext
```

Detta gör att extraktionsmodellen för textsegment kan skilja på året från sökfrågan och året från årsredovisningen. Detta förklaras mer utförligt i extraktionssteget.

4.1.4 Hantering av tabellsegment

För segmentering av tabeller, behandlas tabellerna radvis, så varje rad blir ett separat segment. För varje tabell identifieras kolumnrubriker, spann över kolumnrubriker och enhet (MSEK, mkr eller kr). Positionsbaserad indexering används för varje segment för att identifiera och matcha varje rad och dess värde med tillhörande kolumnrubrik. Om tabellen innehåller ett spann, dvs. en kolumnrubrik som sträcker sig över flera kolumnrubriker, används även positionsbaserad indexering, där spannet blir en del av kolumnrubriker inom parentes, se Figur 5.

	2024			2023		
Skattekostnader (Mkr)	Resultat-räkningen	Övrigt totalresultat	Totalt skatt	Resultat-räkningen	Övrigt totalresultat	Totalt skatt
Aktuell skattekostnad	−2 077	—	−2 077	−2 373	—	−2 373
Uppskjuten skatt	−125	−150	−275	−31	83	52
Totalt	**−2 202**	**−150**	**−2 352**	**−2 404**	**83**	**−2 321**

Figur 5: *Skärmavbild från Skf årsredovisning 2024 för tabell med ett spann*

För exemplet i Figur 5 skapas kolumnrubrikerna:

- Resultaträkningen (2024)
- Övrigt totalresultat (2024)
- Totalt skatt (2024)
- Resultaträkningen (2023)
- Övrigt totalresultat (2023)
- Totalt skatt (2023)

Denna hantering av spann görs för att minska risken att information går förlorad. För radsegment där vissa kolumnvärden saknas, placeras en pseudorepresentation för värdet med tecknet ”-”. Detta görs för att få ett jämnt antal värden för varje kolumnrubrik.

För att skapa en så effektiv extraktion som möjligt, skapas ett format på extraktionstexten som underlättar regelbaserad extraktion. Extraktionstexten har därför följande format med tydliga separationer mellan varje del.

```
TABELL: Tabellnamn | RAD: KPI | VÄRDE_Kolumn1: Värde 1 | VÄRDE_Kolumn2: Värde 2 |
... | VÄRDE_KolumnN: Värde N | ENHET: Enhet | NIVÅ: Nivå
```

Eftersom tabeller kan ha flera olika format görs en regelbaserad lösning för att sortera och hantera dessa olika varianter. För tabeller med klara år (enbart år) i kolumnhuvudet används kolumninformationen direkt.

```
Kolumner: 2023, 2024
↓
TABELL: Koncernens balansräkning | RAD: Summa tillgångar | VÄRDE_2023: 15 |
VÄRDE_2024: 38 | ENHET: msek | NIVÅ: Koncern
```

I vissa tabeller presenteras året som ett spann, se Figur 5. Dessa formateras som:

```
Kolumner: Kortfristiga skulder (2024), långfristiga skulder (2024), Kortfristiga
                    skulder (2023), långfristiga skulder (2023)
                                        ↓
TABELL: Totalresultat över skulder | RAD: Summa skulder | VÄRDE_2024 (Kortfristiga
  skulder): 3 | VÄRDE_2024 (långfristiga skulder): 21 | VÄRDE_2023 (Kortfristiga
 skulder): 7 | VÄRDE_2023 (långfristiga skulder): 18 | ENHET: msek | NIVÅ: Koncern
```

För fall då kolumnrubrikerna innehåller varierad mängd år eller inga år, används kolumnerna direkt likt kategorin enbart år. För tabellsegment undersöks två typer av retrievaltexter för att undersöka hur textens utformning påverkar retrievalresultatet:

```
Strukturerat: KPI. KPI. (Synonym1, Synonym2,..., SynonymN). Nivå. Tabelltyp. År.
                                        Enhet
Querybaserad: Vad är KPI, (Synonym1, Synonym2,..., SynonymN)? Nivå. Tabelltyp. År.
                                        Enhet
```

4.1.5 Metadata-förstärkt representation

Utöver den textuella representationen och dess embedding sparas också viktig metadata. Detta är attribut som beskriver segmentets position och funktion i dokumentets hierarki. Syftet med detta är att möjliggöra kontextuell filtrering, exakt numerisk extraktion samt hybrid retrieval där semantisk närhet kombineras med strukturell precision.
Typer av metadata:

- Dokument-ID: ID för vilket segmentet tillhör.
- Segment-ID: Unikt ID för varje segment i dokumentet.
- Segmenttyp: Beskriver om segmentet är av typen text eller tabell.
- Sidnummer: Anger vilken sida segmentet befinner sig på.
- Sektion: Härledd via regelbaserad analys av rubriker. Exempelvis noter, förvaltningsberättelse, finansiella rapporter etc. Sektionen utgör en överordnad dokumentklassificering och möjliggör strukturell filtrering.
- Rubrik: Den lokala rubriken som segmentet ligger under.
- Enhet: Om enheten identifieras inom segmentet, annars sätts okänd.
- Nivå: Koncern eller moderbolag. Detta är särskilt viktigt i finansiella rapporter där samma KPI kan förekomma för både koncern och moderbolag.
- Källtyp: Beskriver vad för typ källan är. Detta är viktigt då en primärkälla ses som viktigare informationskälla än en sekundärkälla. Exempelvis Primär tabell, Sekundär tabell, Not tabell, Not text osv.
- Tabelltyp: Anges för tabellsegment för att identifiera primära tabeller som resultaträkningen, balansräkningen och kassaflödesanalysen.

4.2 Embedding & Indexering

För att möjliggöra identifiering och jämförelse mellan segment och sökfråga i retrievalsteget, skapas en embeddingrepresentation av retrievaltexten med hjälp av en transformermodell. För att underlätta en effektiv sökning i retrievalsteget behöver segmenten och dess embeddingrepresentation lagras på korrekt sätt i en vektordatabas.

4.2.1 Embeddingmodell

Modellerna som används i denna studie är Multilingual-E5-large och två konfigurationer av BGE-M3, en med enbart en dense-vektor och en med både sparse- och dense-vektor. Valet av dessa modeller grundar sig i deras etablerade användning i moderna retrieval-baserade Natural Language Processing (NLP) system samt rapporterat resultat i tidigare studier inom semantisk sökning och RAG-liknande system [2] [13].

Modellerna är även öppet tillgängliga, vilket möjliggör lokal körning utan beroende av externa API-tjänster. Detta är särskilt viktigt för reproducerbarhet samt praktisk implementation av systemet.

4.2.2 Vektorbas och indexering

För att kunna identifiera relevanta segment utifrån en sökfråga används vektordatabasen Milvus. Varje årsredovisning skapar en egen separat kollektion där tillhörande segment lagras. Varje kollektion följer ett schema, som innehåller index, metadatafält, embeddingvektorer etc.

Retrievaltexten embeddas med respektive modeller och skapar en embeddingvektor (dense-vektor) för semantisk sökning, som sedan normaliseras för bättre jämförelser mellan semantisk betydelse i vektorrummet. För BGE-M3 med både sparse- och dense-vektor skapas även en sparse-vektor.

Varje segment i samma dokument indexeras i Milvus under sin kollektion, genom ett förberett schema, med data (retrievaltext och extraktionstext), metadata och embeddingvektorer. Då olika modeller har olika embeddingdimensioner behöver Milvus konfigureras med motsvarande dimension. För fulla implementationsdetaljer se Appendix B.

Inner Produkt (IP) används som ett utvärderingsmått i Milvus vektormatchning. Valet av IP beror på att moderna modeller som BGE-M3 och Multilingual-E5-large är tränade för semantisk likhet baserad på vektorernas skalärprodukt. Genom att använda samma likhetsmått vid sökning som modellerna optimerats för under träning kan retrievalsteget bättre identifiera semantiskt relevanta segment.

4.3 Retrieval

För att kunna extrahera korrekt information utifrån en sökfråga, behöver korrekt segment identifieras från årsredovisningen. Retrievalsteget utgår från att identifiera och inhämta relevanta segment utifrån ställd sökfråga. Detta steg är uppdelat i två delar. Den första delen utgår från en hybridsökningsmodell för att identifiera relevanta segment i Milvus. Nästa del är en rerankmodell som försöker omarrangera och komplettera segmentens ordning utifrån dess metadata. Steget avslutas med att skapa en kandidatpool av top-10 segment från respektive segmenttyp (tabellsegment, textsegment).

4.3.1 Hybridsökning

För att komplettera styrkorna av både semantisk- och lexikal sökning används en hybridsökningsmodell. Detta betyder att flera olika perspektiv söks samtidigt och sedan sammanfogas för en gemensam bedömning av segmentens position i sökningen. Detta görs för att förbättra sökningsmöjligheten och täcka upp för respektive metods svagheter. Varje sökning sätter ett hårt filter på metadatanivå, där en exkludering av segment på moderbolagsnivå görs. Detta beror på att studiens fokus ligger på koncernen. Innan någon sökning eller embedding påbörjas görs en query-extension, där KPI-synonymer

läggs till i frågan. Detta görs för att fånga upp fall då frågans synonyma KPI använts i årsredovisningen. Sökfrågan embeddas med samma modell som användes för att generera embeddings i Milvus.

I denna pipeline används tre olika sökmetoder. Den första sökningen är en ren semantisk sökning. Denna sökning identifierar semantiska likheter, synonymer och omformuleringar mellan segmenten och respektive fråga. Den utgår från en embedding på hela frågan med respektive synonymer.

Den andra sökmetoden är en lexikal sökning. Genom att fokusera på exakta ord och dess vikt i segmentet, identifieras många segment som lätt hade kunnat missats eller fått en lägre poäng i en ren semantisk sökning.

Den tredje sökningsmetoden är en semantisk sökning på en förkortad variant av frågan där endast sökfrågans ”KPI” används. Det vill säga frågan ”Vad är rörelseresultat, 2024?” omformuleras till endast ”rörelseresultat”. Detta görs för att stärka signalen och styrka tyngden av KPI:ns betydelse i kontexten.

Varje sökning resulterar i en separat resultatlista. För att kombinera dessa till en och samma lista appliceras en Reciprocal Rank Fusion (RRF). För varje segment summeras dess RRF-poäng baserat på dess rankposition i respektive lista, vilket resulterar i en gemensam slutranking.

En individuell hybridsökning appliceras på respektive segmenttyp. Detta görs på grund av att tabellsegmenten i stor utsträckning är mer koncentrerade, kortare och innehåller mer specifik information. Därför poängsätts tabeller ofta högre i sökningen än text som vanligtvis innehåller utspridd information med varierande ämnen. För textsegment blir embeddingvektorn därmed ett ”genomsnitt” av textens betydelse, vilket gör att hybridsökningsresultatet mer partisk mot tabellsegmenten.

Milvus tar in två parametrar vid sökning i vektordatabasen, en embeddad sökfråga och ett antal segment som ska hämtas (k), så bestäms två separata k-värden, en för varje segmenttyp. Resultaten för beräkning av varje k värde kan utläsas i Appendix C

4.3.2 Reranking

En rerankingmetod används för att förbättra retrievalkvaliteten och finjustera rangordningen av segment baserat på innehåll och metadata. Rerankingen sker på två plan, en lexikal rerank och en metadataboost.

Den lexikala rerankingen går ut på att lexikalt identifiera likheter mellan ord och meningar i segmentets kontext utifrån efterfrågat KPI och ge positiv eller negativ viktning beroende på likhetspoäng. För att göra det används Rapidfuzz, ett Python bibliotek för fuzzy matching [29].

För tabellsegment används heuristiska regler i kombination med fuzzy matching för att identifiera likheter mellan segmentets KPI och frågans KPI. Det inkluderar tokenbaserad fuzzy matching, begreppstäckning genom tokenöverlapp, teckenbaserad likhet samt en särskild jämförelse där mellanslag ignoreras för att hantera sammanskrivna ord som kan uppstå vid dokumentkonvertering.

För textsegment används istället partiell fuzzy matching i kombination med begreppstäckning. Detta eftersom KPI:n i många fall endast utgör en mindre del av ett längre textstycke.

Nästa steg i rerankingen är metadata boost. Denna del ger positiv eller negativ viktning till segment, beroende på dess metadata. Viktningen utgår från källtyp, nivå och enhet. En extra belöning ges till tabellsegment där tabelltypen är av typen resultaträkning, balansräkning eller kassaflödesanalys. Detta då majoriteten av KPI återfinns i dessa.

En cross-encoder reranker utvärderades även som ett alternativ till den heuristiska rerankingen. Modellen som används är bge-reranker-large, en cross-encoder-reranker som ofta används inom RAG-system.

Fullständiga implementationsdetaljer kan utläsas i Appendix D.

4.3.3 Retrieval Utvärdering

För att utvärdera retrievalstegets prestation används ett flertal mått. MRR, primära först, recall@k, primär källa@k samt precision@20 och recall@20 för korrekta respektive relevanta källor.

En relevant källa i denna studie avser ett segment som innehåller det efterfrågade KPI:t och som kan bidra till att besvara frågan. Beroende på innehållstyp kan den relevanta källan utgöras av ett tabellsegment för tabellbaserade värden, eller ett textsegment där KPI uttrycks explicit för textbaserat innehåll.

Recall används för att mäta hur ofta en relevant källa återfinns bland de k hämtade kandidaterna. Måttet utvärderas genom att jämföra top-k segment med relevanta segment angivna i goldsetet. Detta genomförs genom att jämföra segment-ID mellan hämtade segment och goldset.

MRR används för att utvärdera hur högt upp i rankningen den första relevanta källan återfinns. Måttet *primär först* används för att beskriva andelen då den primära källan placeras först.

Måttet *primär källa@20* beskriver hur ofta den fördefinierade primära källan återfinns bland de 20 högst rankade segmenten. Detta mått används för att undersöka systemets förmåga att identifiera den primära källan.

Precision@20 (korrekta källor) och recall@20 (relevanta källor), används för att ge en mer detaljerad bild av kvaliteten i top-20 listan. Precision@20 mäter andelen relevanta segment som även innehåller korrekt värde värde i top-20, medan recall@20 (relevanta källor) mäter andelen relevanta källor som återfinns inom samma mängd.

4.4 Extraktion

Extraktionen utgår från att, givet de segment som hämtats från tidigare steg, extrahera värdet i respektive segment. För tabellsegment som är mer strukturerande kan en heuristik extraktion göras. För textsegment med mer språklig variation krävs en djupare förståelse för den semantiska betydelsen i texten, vilket motiverar valet av en QA-modell.

4.4.1 Tabellsegment

För tabellsegment används en regelbaserad metod som kombinerar textmatchning, reguljära uttryck och fuzzy matching. Metodens indata består av två delar: en sökfråga samt ett kontextfält som representerar en tabellrad i textformat.

```
                    Vad är KPI, ÅR?
                          ↓
TABELL: Tabelltyp | RAD: KPI | VÄRDE_Col1: Värde1 | VÄRDE_Col2: Värde2 | ... |
          VÄRDE_ColN: VärdeN | ENHET: Enhet | NIVÅ: Nivå
```

Det efterfrågade KPI:t och aktuella året extraheras utifrån sökfrågan med hjälp av reguljära uttryck som fångar mönster av kolumninformation och respektive värde.

Därefter analyseras tabellens kontext. Kontexten är strukturerad med särskilda etiketter som indikerar rader och värden. Först identifieras det relevanta KPI:t i tabellen genom att extrahera text efter etiketten ”RAD”. Sedan extraheras alla förekomster av värden i tabellen tillsammans med tillhörande kolumninformation och skapar en *kontextinformation*. Kontextinformationen består av tre delar, en kolumnetikett, en kvalificerare och ett värde. Detta kan observeras i nedanstående exempel.

```
TABELL: Koncernens resultaträkning | RAD: Rörelseresultat | VÄRDE_2023 (Koncern):
    34 | VÄRDE_2024 (Koncern): 23 | VÄRDE_2023 (Moderbolag): 75 | VÄRDE_2024
              (Moderbolag): 56 | ENHET: msek | NIVÅ: Koncern
                                    ↓
                              Kontextinformation:
   [{2023, Koncern, 34}, {2024, Koncern, 23}, {2023, Moderbolag, 75}, {2024,
                                Moderbolag, 56}]
```

Efter att kontextinformation har extraherats sker en urvalsprocess för att identifiera det mest sannolika svaret och generera kandidater. Om kontextinformationen innehåller explicita årtal filtreras kandidaterna direkt baserat på det år som efterfrågas i sökfrågan.

```
                          Vad är rörelseresultat, 2024?
                                    ↓
  TABELL: Koncernens resultaträkning | RAD: Rörelseresultat | VÄRDE_2023: 34 |
               VÄRDE_2024: 23 | ENHET: msek | NIVÅ: Koncern
                                    ↓
                              Kontextinformation:
                           [{2023, , 34}, {2024, , 23}]
                                    ↓
                                Kandidater:
                                    ↓
                              [{2024, , 23}]
```

Om årtal saknas eller inte är tydligt kopplade till ett värde används istället fuzzy matching för att jämföra likheten mellan det efterfrågade KPI:t och tillgängliga etiketter i tabellen i olika steg. Första steget jämför radens KPI med sökfrågans KPI, där en likhetspoäng som överstiger 90 anses som ett relevant segment att undersöka, där samtliga värden tas in som kandidater.

```
                          Vad är rörelseresultat, 2024?
                                    ↓
 TABELL: Totalresultat över året | RAD: Rörelseresultat | VÄRDE_Koncernen: 34 |
            VÄRDE_Moderbolaget: 75 | ENHET: msek | NIVÅ: Koncern
                                    ↓
                              Kontextinformation:
                 [{Koncernen, "", 34}, {Moderbolaget, "", 75}]
                                    ↓
                                Kandidater:
                                    ↓
                 [{Koncernen, "", 34}, {Moderbolaget, "", 75}]
```

För fall då likhetspoängen resulterar i ett värde lägre än 90, används fuzzy matching för att undersöka kolumnrubrikers likhet med sökfrågans KPI. Detta då en radvis segmentering kan ge fall där kolumnerna i tabellen innehåller sökfrågans KPI men raden har en mer generell term som ”Summa” eller ”Totalt”. Ett tröskelvärde på 80 definieras och kolumner i tabeller går igenom en fuzzy matching för att undersöka deras likhet till sökfrågans KPI. Fall som överstiger tröskelvärdet tas in som kandidater.

```
                          Vad är rörelseresultat, 2024?
                                    ↓
```

```
TABELL: Totalresultat över året | RAD: Totalt | VÄRDE_Eget kapital (Koncernen): 17
 | VÄRDE_Eget kapital (Moderbolaget): 8 | VÄRDE_Rörelseresultat (Koncernen): 34 |
      VÄRDE_Rörelseresultat (Moderbolaget): 75 | ENHET: msek | NIVÅ: Koncern
                                        ↓
                                Kontextinformation:
[{Eget kapital, Koncernen, 17}, {Eget kapital, Moderbolaget, 8}, {Rörelseresultat,
               Koncernen, 34}, {Rörelseresultat, Moderbolaget, 75}]
                                        ↓
                                    Kandidater:
                                        ↓
     [{Rörelseresultat, Koncernen, 34}, {Rörelseresultat, Moderbolaget, 75}]
```

Om kandidatgenereringen enbart producerar en kandidat väljs detta värde direkt. För fall där flera kandidater återstår används en prioriteringsmekanism för att välja det mest relevanta värdet. Denna mekanism bygger på en fördefinierad prioriteringslista av nyckelord (Exempelvis ”koncern”, ”total”, ”summa”) som tilldelas olika prioriteringspoäng. Kandidater med högre prioritet (lägre poäng) föredras. Slutligen genomgår det predikterade värdet en normaliseringsprocess där tecken som parenteser, mellanslag och punkter tas bort för att möjliggöra korrekt jämförelse med det faktiska svaret.

Den fulla prioriteringslistan kan utläsas i Appendix E.

4.4.2 Textsegment

Extraktionen för textbaserade segment använder sig av en finjusterad transformer QA (Question Answering) modell. Modellen används för att identifiera det mest sannolika svarsspannet i kontexten baserat på en sökfråga.

Datasetet består av textsegment (kontext), tillhörande sökfrågor samt annoterade svar med angivna startpositioner i texten. För att möjliggöra modellträning tokeniseras både sökfrågor och kontext med en förtränad tokenizer.

Under tokeniseringen beräknas även kopplingen mellan teckenpositioner i originaltexten och motsvarande tokenpositioner. För att koppla de annoterade svaren till modellens tokenrepresentation används en offset mapping, vilket översätter de annoterade svarens start- och slutpositioner till motsvarande tokenindex. Detta är nödvändigt för att träna modellen.

Modellen som används är XLM-RoBERTa-base-squad2, en flerspråkig transformerbaserad QA-modell, baserad på arkitekturen XLM-RoBERTa. Modellen predikterar sannolikhetsfördelningen över start- och slutpositioner i texten.

Finjustering av modellen optimeras genom att minimera skillnaden mellan predikterade och sanna spannpositioner. Finjustering sker över flera epoker med minibatch-gradientnedstigning, och inkluderar tekniker såsom early stopping för att förhindra överanpassning. Efter varje epok utvärderades modellen på valideringsmängden, där F1-score används som kriterium för att välja den bästa modellen. Anledningen till valet av F1-score är att den bättre speglar faktisk svarskvalitet (tokenöverlappning) än enbart loss på start- och slutpositioner.

Vid inferens tokeniseras sökfrågan och textens kontext. Modellen används för att prediktera start- och slutpositioner för det mest sannolika svaret. Dessa positioner konverteras sedan tillbaka till text genom avkodning av motsvarande tokenintervall.

Slutligen genomgår det predikterade värdet en normaliseringsprocess där tecken som parenteser, mellanslag och punkter, tas bort för att möjliggöra korrekt jämförelse med det faktiska svaret.

4.4.3 Extraktion utvärdering

För att utvärdera metodernas prestanda används två standardmått: Exact Match (EM) och F1-score, definerade i Ekvation 3 och 6. EM mäter andelen fall där det predikterade svaret exakt överensstämmer med facit. F1-score mäter tokenbaserad överlappning, där precision och recall beräknas utifrån gemensamma tokens mellan de två strängarna.

Extraktionen utvärderas både separat för de olika segmenttyperna samt genom en sammantagen bedömning för att analysera respektive komponents bidrag till den totala extraktionsförmågan. För tabellsegment utvärderas även olika kontextinformationer som enbart år, blandade/år i spann eller inga år.

4.5 Triangulering & Verifiering

I denna studie används en triangulering för att öka trovärdigheten av slutresultatet. Från extraktionssteget identifieras det mest sannolika svaret utifrån ett enskilt segment. Detta innebär att möjligheten att jämföra information från flera oberoende källor och därigenom utvärdera resultatets korrekthet saknas. Trianguleringen används därför som ett sätt att utvärdera extraherade värden från enskilda segment i extraktionen och sammanväga dessa för att bedöma vilket resultat som har starkast evidens. Detta möjliggör både validering av överensstämmande källor och identifiering av motstridiga resultat.

Trianguleringen utförs genom en aggregering av flera oberoende evidenskällor från dokumentet. Denna metod har uppvisat goda resultat i tidigare studier [61]. Studiens trianguleringsmetod är baserad på en heuristisk kombination av konsistens, källtillförlitlighet, kontextmatchning och rimlighetskontroller.

Trianguleringen bygger på tre huvudsakliga informationskällor, retrievalresultatet, metadata samt extraherade värden från extraktionssteget. Segment där inget numeriskt värde kunde extraheras exkluderas från vidare beräkning, men resterande kandidater behålls för trianguleringen.

Varje kandidat tilldelas en viktad poäng baserat på följande faktorer:

- Källtyp (exempelvis tabell, text eller not),
- Tabelltyp i förhållande till KPI,
- Hierarkisk nivå (t.ex. koncern eller moderbolag),
- Semantisk likhet från retrievalsteget.

För tabellsegment används även en tabelltyp-mappning till respektive primära tabell (balansräkning, resultaträkning och kassaflödesanalys), vilket belönar överensstämmelser mellan respektive KPI och förväntad finansiell rapportstruktur. Exempelvis förväntas ”Totala tillgångar” återfinnas i balansräkningen. Full specifikation kan utläsas i Appendix F.

Varje segment grupperas därefter baserat på deras extraherade värde och numeriska likhet, med en relativ tolerans på 1% för att hantera avrundningsskillnader. Poängen inom varje grupp summeras varpå gruppen med högst poäng blir den vinnande kandidaten och tillhörande värdet returneras som slutgiltigt svar.

För trianguleringen beräknas även ett tillförlitlighetsmått baserat på poängfördelningen mellan grupperna:

- **Konsensus**: Samtliga kandidater tillhör samma grupp.
- **Klar vinnare**: En grupp dominerar tydligt i poäng.

- **Mjuk vinnare**: Mindre skillnad mellan toppgrupper.
- **Konflikt**: Flera grupper har konkurrerande poäng.

4.5.1 Triangulering utvärdering

Trianguleringen utvärderas med EM (Exact Match) och F1-score på systemnivå genom att jämföra det slutgiltiga värdet från triangulering med det annoterade goldset-värdet. Detta ger ett mått på hela pipelinens prestation, vilket skiljer sig från extraktionssteget som endast mäter kvaliteten av extraktion i enskilda fall.

Trianguleringens utvärdering fångar därmed den kombinerade effekten av retrieval, extraktion och triangulering, snarare än isolerade delkomponenter.

Skillnader mellan konfigurationer i trianguleringsutfallet analyseras för att identifiera hur känslig beslutslogiken är för variationer i tidigare pipeline-steg.

4.6 Avgränsningar

Denna studie innehåller ett flertal metodologiska avgränsningar som påverkar systemets generaliserbarhet och tolkning av resultaten.

Studien utgår endast från svenska årsredovisningar i PDF-format och KPI-relaterade frågor. Detta innebär att modellen inte är uppbyggd eller utvärderad på generella question-answering-scenarier utanför den finansiella domänen. Skannade PDF-dokument eller andra dokument som kräver OCR-baserad extraktion inkluderas inte i studien. Ännu en avgränsning är att bilder och figurer i årsredovisningarna exkluderas, då fokus ligger på extraktionsförmågan från text och tabeller.

Studien fokuserar på faktabaserad, numerisk informationsutvinning snarare än tolkande frågor, såsom förklaringar till förändringar i exempelvis rörelseresultat eller beskrivningar av bolagets strategi. Trianguleringen avgränsas till deterministisk poängsättning, snarare än generativa resonemang från språkmodeller.

Segmenteringen baseras även på en automatiserad dokumentkonvertering med Docling, där ingen manuell korrigering av strukturell metadata utförs. Detta innebär att systemet är beroende av att den extraherade dokumentstrukturen är korrekt.

En annan avgränsning som görs är att segmenteringen bygger på en adaptiv, radbaserad metod för tabeller. Detta innebär att tabellers fullständiga strukturella form inte alltid bevaras, vilket kan leda till informationsförluster i fall där rad/kolumn relationer är mer komplexa.

Retrievalsteget avgränsas till att begränsa antalet top-k segment per fråga till ett fast antal (20), och ingen dynamisk anpassning för olika retrievaldjup appliceras. Retrievalsteget baseras på BGE-M3 med en kombination av både sparse- och dense-representationer, tillsammans med en heuristisk rerankingsmetod.

Extraktionsteget avgränsas med att enbart undersöka ett val av extraktionsmetod per segmenttyp.

En annan avgränsning som görs är att trianguleringsteget utgår från en enda heuristisk modell och baseras på en fast viktning av olika signaler som KPI-likhet, källtyp, tabelltyp och nivå. Detta innebär att ingen inlärd modell eller probabilistisk osäkerhetsmodell används. Vidare begränsas beslutsprocessen till en enkel värdegruppering med en relativ toleransnivå, samt fyra diskreta tillförlitlighetsklasser (konsensus, klar vinnare, mjuk vinnare och konflikt).

Slutligen returnerar systemet ett enskilt numeriskt värde per fråga utan statistiska konfidensintervall eller explicit osäkerhetskalibrering.

4.7 Goldset

I denna studie används tre olika facit (goldset). Det första är ett tränings-goldset, vilket används för att bygga och utvärdera pipelinens olika steg.

Därefter skapas två olika test-goldset genom en fullständig annotering av samtliga KPI:er. Det första goldsetet tar hänsyn till fel som uppstått i dokumentkonvertering och facit anpassas därefter. Det handlar om källor som saknas, tabellvärden som blivit förändrade samt tabellstrukturer som förändrats. Detta goldset används för utvärdering av pipelinens prestation. Det andra test-goldset behåller de fel som uppstått vid dokumentkonverteringen. Detta görs för att kunna skapa en jämförelse och utvärdera dokumentkonverteringens påverkan på resultatet.

Fördelningen av företag, årsredovisningar, KPI:er och datapunkter för respektive goldset presenteras i Tabell 1.

Tabell 1: *Fördelning av årsredovisningar, KPI och datapunkter för respektive goldset.*

Goldset	Antal företag	Antal årsredovisningar	Antal KPI	Antal datapunkter
Träning	13	29	36	135
Test (utan Doclingfel)	5	15	44	705
Test (med Doclingfel)	5	15	44	735

Varje datapunkt i respektive goldset innehåller följande delar:

1) Ett årsredovisningsdokument
2) Ett specifikt KPI
3) Ett specifikt år
4) En KPI-fråga för specifikt år
5) En given nivå (koncern, moderbolag)
6) En given källtyp (primär tabell, nottext etc.)
7) Svarsalternativ för datapunkten (Primär källa samt sekundära källor)

Samma dokument kan ge upphov till flera datapunkter.

För varje årsredovisning i träningssetet skapas ett urval utifrån de 36 valda KPI:erna, där fem KPI:er slumpas per årsredovisning, vilket leder till 145 datapunkter. Detta görs för att testa samma KPI på olika årsredovisningar med olika strukturer för att skapa en generalisering. På grund av att vissa KPI:er saknades i årsredovisningen gjordes en sållning, vilket resulterade i 135 slutgiltiga datapunkter.

För varje svarsalternativ i goldset dokumenteras primära källan, vanligtvis finansiella rapporter som balansräkning, resultaträkning eller kassaflödesanalys. Vid specifika KPI:er som ej erhölls i finansiella rapporter, görs ett val av annan rimlig primär källa. Detta val baseras på källans format och struktur, där tabeller prioriteras. För fall då efterfrågad information kan erhållas från flera platser i årsredovisningen dokumenteras även sekundärkällor, med samma tillhörande information som för den primära källan. Detta för att möjliggöra en mer rättvis utvärdering av retrievalsteget, då flera källor kan innehålla relevant och korrekt information trots att de inte utgör den primära källan. Detta möjliggör en uppdelad utvärdering av dels identifieringen av den primära källan samt systemets förmåga att återfinna ytterligare relevanta källor.

För träningssetet definieras relevanta segment som segment som innehåller det efterfrågade KPI:t och det korrekta värdet. Under arbetets gång identifierades att denna definition var för strikt i relation till retrievalstegets uppgift. För de två testseten utökas därför relevansdefinitionen till att inkludera segment som innehöll det efterfrågade KPI:t, men inte nödvändigtvis det korrekta värdet. Syftet med detta är att bättre spegla retrievalsystemets förmåga att identifiera relevant information.

Annoteringen av goldset gjordes manuellt genom granskning av årsredovisningar och kontroll av varje datapunkt mot dess ursprungliga källa.

5 Resultat

Nedan följer resultatet av studien. Resultaten presenteras stegvis i samma flöde som studiens pipeline. Första delen går igenom resultatet för retrievaldelen, följt av extraktionsresultaten och avslutas sedan med trianguleringsresultatet.

5.1 Retrieval

Resultatet för retrievaldelen är uppdelat i följande delar. Först presenteras resultatet av tre olika embeddingmodeller för att identifiera den mest lämpliga modellen för vidare analys. I den andra delen visas resultatet på en fördjupad utvärdering av den valda modellen (BGE-m3 med sparse- och dense-vektor), där effekten av reranking, olika embeddingtexter samt tränings- och testdata undersöks. Även fall då Doclingfel upptäckts utvärderas. Tabell 3 och 5 visar resultat där Doclingfel är tydliga och datarensning har gjorts.

5.1.1 Olika retrievalmodeller

För att jämföra retrievalmodellernas påverkan utvärderades tre modellkonfigurationer.

Tabell 2: *Resultat för olika modellkonfigurationer på träningset utan rerank.*

Typ	n	MRR	Primär först	Recall@1	Recall@3	Recall@5	Recall@10	Recall@20
Multilingual-e5-large	20	0.69320	0.51852	0.57778	0.78519	0.84444	0.90370	0.91852
BGE-m3 (Dense)	20	0.74990	0.50370	0.64444	0.80000	0.87407	0.94815	0.96296
BGE-m3 (Dense + Sparse)	20	0.72580	0.40000	0.59259	0.82963	0.90370	0.96296	0.97778

Tabell 2 visar resultatet på tränings-goldset utan någon rerank för de tre embeddingmodellerna Multilingual-e5-large, BGE-M3 (dense) samt BGE-M3 (sparse + dense). BGE-M3 modellerna uppvisar högre MRR-värden än E5, där BGE-M3 (dense) uppnår högst värde på 0.74990 och BGE-M3 (sparse + dense) uppnår 0.72580, jämfört med 0.69320 för E5. Detta betyder att de två BGE-modellerna vanligtvis rankar den första relevanta källan högre upp än E5.

För mätvärdet primär först visar E5 och BGE-M3 (dense) liknande resultat på 0.51852 och 0.50370 respektive. BGE-M3 (sparse + dense) visar ett primär först värde på 0.40000.

För recall måtten observeras generellt högre värden för båda BGE-M3 modellerna jämfört med E5. BGE-M3 (sparse + dense) uppvisar högsta värden när det kommer till Recall@3, Recall@5, Recall@10 och Recall@20. BGE-M3 (dense) visar högsta värdet på Recall@1 på 0.64444 jämfört med BGE-M3 (sparse + dense) på 0.59259.

Sammantaget visar resultatet att BGE-M3 med en kombinerad sparse- och dense representation ger bäst resultat för högre recall@k värden medan dense-variationen preserar bäst vid top-rank precision. Eftersom studiens pipeline använder triangulering för att verifiera det slutligt värdet anses recall@20 vara det mest relevanta måttet, då det är viktigare att rätt segment identifieras än vilken position det har i rankningen.

5.1.2 Retrievalresultat på olika konfigurationer och goldets

Samtliga resultat i detta avsnitt avser endast BGE-M3 modellen med kombinerad sparse- och dense representation. Resultaten avser även en kombinerad retrieval av både text- och tabellsegment.

Tabell 3: *Retrievalresultat på tränings-goldset.*

Konfiguration	MRR	Primär först	Recall@1	Recall@3	Recall@5	Recall@10	Recall@20
Strukturerad text	0.72580	0.4	0.59259	0.82963	0.90370	0.96296	0.97778
Strukturerad text + rerank	0.85365	0.72593	0.78518	0.91111	0.91111	0.97037	0.97037
Querybaserad text	0.81312	0.54815	0.74074	0.87407	0.91111	0.96296	0.97778
Querybaserad text + rerank	0.86096	0.74815	0.8	0.91111	0.91111	0.97037	0.97037

Tabell 3 visar resultatet på tränings-goldsetet för fyra konfigurationer: med och utan reranking samt strukturerad och querybaserad text.

Resultaten visar en genomgående förbättring av MRR och recall@1-3 vid användning av rerank. För strukturerad text kan en ökning av MRR från 0.72580 till 0.85365 observeras, medan motsvarande ökning för querybaserad text är från 0.81312 till 0.86096. Även i fall då den primära källan rankas först förbättras respektive mått för båda konfigurationerna vid användning av rerank. Detta innebär att rerank förbättrar ordningen högre upp i resultatlistan.

Querybaserad text ger generellt bättre resultat än strukturerad text i konfigurationer där rerank inte används. Detta observeras särskilt för MRR, med ett värde på 0.81312 och 0.72580 för respektive konfiguration samt recall@1, med ett värde på 0.74074 och 0.59259. Då reranking introduceras, minskas dessa skillnader och liknande nivåer av värden kan observeras i samtliga punkter.

För recall @5-10 uppvisas små variationer mellan de olika konfigurationerna, men ingen tydlig skillnad visas vid användning av varken reranking eller querybaserad text.

Tabell 4: *Retrievalresultat på träningsset för studiespecifika mått.*

Konfiguration	Primär källa@20	Precision@20 (korrekta källor)	Recall@20 (relevanta källor)
Strukturerad text	0.90370	0.66667	0.68382
Strukturerad text + rerank	0.95556	0.64706	0.65931
Querybaserad text	0.88148	0.65686	0.67402
Querybaserad text + rerank	0.95556	0.64461	0.65686

Resultaten i Tabell 4 visar att samtliga konfigurationer uppnår en hög andel primär källa@20, där värdena ligger mellan cirka 0.88 och 0.96. Detta tyder på att retrievalsteget lyckas identifiera och inkludera den huvudsakliga primära källan i de flesta fall inom de top-20 returnerade segmenten.

Precision@20 visar relativt små variationer mellan konfigurationerna, där strukturerad- och querybaserad text presterar marginellt bättre än motsvarande reranking konfigurationerna. Dessa variationer är dock begränsade och tyder inte på någon större förvärring av precision vid användning av rerank.

Recall@20 visar en stabil nivå över samtliga konfigurationer, med små variationer mellan cirka 0.65 och 0.68. Detta visar att den initiala retrievalkomponenten har en relativt hög täckning av relevanta källor utan något beroende på rerank.

Sammantaget visar resultaten att retrievalsystemet i grund skapar en stark kandidatgenerering, oberoende av rerank, där huvuddelen av relevanta och korrekt källor återfinns inom top-20.

Tabell 5: *Retrievalresultat på test-goldset.*

Konfiguration	MRR	Primär först	Recall@1	Recall@3	Recall@5	Recall@10	Recall@20
Strukturerad text	0.74981	0.33191	0.64255	0.84823	0.89362	0.94043	0.94752
Strukturerad text + rerank	0.87728	0.68652	0.82128	0.92340	0.93617	0.96312	0.96738
Querybaserad text	0.80791	0.41560	0.73475	0.87376	0.91064	0.95035	0.95745
Querybaserad text + rerank	0.87281	0.68794	0.81135	0.92482	0.93617	0.96596	0.97021

Tabell 5 visar resultat på test-goldsetet för fyra konfigurationer: med och utan reranking samt strukturerad text och querybaserad text.

Resultaten visar en genomgående förbättring av samtliga mätpunkter vid användning av reranking. För MRR kan en ökning observeras från 0.74981 till 0.87728 för strukturerad text samt 0.80791 till 0.87281 för querybaserad text. En högre ranking av primära källan kan även identifieras vid användning av reranking. För strukturerad text går mätvärdet från 0.33191 till 0.68652 samt från 0.41560 till 0.68794 för querybaserad text. Detta innebär att även för testsetet ger reranking ett förbättrat resultat för de högst rankade segmenten.

Querybaserad text ger generellt ett bättre resultat än strukturerad text i konfigurationer där rerank inte används. Skillnaden är särskild tydligt för MRR med 0.74981 och 0.80791, primära först med 0.33191 och 0.41560 samt recall@1 med 0.64255 och 0.73475. Detta innebär att relevanta kandidater rankas högre initialt vid användning av querybaserad text jämfört med strukturerad test. När rerank introduceras minskar dessa skillnader och liknande nivåer observeras för samtliga punkter.

För recall@3-20 uppvisas en viss variation mellan de olika konfigurationerna, där den största skillnaden uppstår på de högst rankade segmenten. Dessa variationer minskar när fler segment introduceras.

Tabell 6: *Retrievalresultat på test-goldset för studiespecifika mått.*

Konfiguration	Primär källa@20	Precision@20 (korrekta källor)	Recall@20 (relevanta källor)
Strukturerad	0.82270	0.44798	0.71964
Strukturerad + rerank	0.92624	0.46747	0.74993
Querybaserad text	0.84397	0.45307	0.72324
Querybaserad text + rerank	0.93050	0.46717	0.78082

Resultaten i Tabell 6 visar att samtliga konfigurationer uppnår en hög andel primär källa@20, där värdena ligger mellan cirka 0.82 och 0.93. Detta tyder på att retrievalsteget lyckas identifiera och inkludera den huvudsakliga primära källan i de flesta fall inom de top-20 returnerade segmenten.

Precision@20 visar relativt små variationer mellan konfigurationerna, där rerankbaserade konfigurationer presterar marginellt bättre än motsvarande konfigurationerna. Denna variation är dock begränsad och tyder inte på någon större förbättring av precision vid användning av rerank.

För recall@20 observeras en liknande trend där konfigurationerna med rerank visar en viss förbättring jämfört med motsvarande konfigurationer utan rerank, vilket är

särskilt tydligt för querybaserad text. Detta visar att rerankingen delvis bidrar till att ranka relevanta källor högre i listan, dock visar konfigurationerna redan innan rerank en stark förmåga att identifiera relevanta kandidater.

Sammantaget visar resultaten att retrievalsystemet i grunden skapar en stark kandidatgenerering, oberoende av rerank, där merparten av relevanta och korrekt källor återfinns inom top-20.

Tabell 7: *Retrievalresultat på test-goldset - med Doclingfel.*

Konfiguration	MRR	Primär först	Recall@1	Recall@3	Recall@5	Recall@10	Recall@20
Querybaserad text	0.78967	0.40136	0.71837	0.85442	0.88980	0.92653	0.93333
Querybaserad text + rerank	0.84354	0.65850	0.77959	0.89660	0.91293	0.94150	0.94558

Tabell 7 visar retrievalresultatet på test-goldset med Doclingfel, vilket utgår från det querybaserade textformatet. Tabellen visar resultatet för både med och utan rerank.

Resultatet visar att samtliga mätvärden ökar vid användning av rerank. Störst skillnad observeras för primära först som förbättras från 0.40136 till 0.65850. Tydliga förbättringar kan även observeras för recall@1-3 och MRR. Recall@5-20 visar även mindre men konsekventa förbättringar, vilket indikerar en generell förbättring i rankingkvalitet i hela retrievalsteget vid användning av rerank.

Tabell 8: *Jämförelse mellan cross-encoder (BGE-reranker-large) och studiens rerank-metod.*

Konfiguration	MRR	Primär först	Recall@1	Recall@3	Recall@5	Recall@10	Recall@20
Querybaserad text	0.81312	0.54815	0.74074	0.87407	0.91111	0.96296	0.97778
Querybaserad text + Cross-encoder	0.78291	0.55556	0.71852	0.82963	0.87407	0.91852	0.92593
Querybaserad text + rerank	0.86096	0.74815	0.8	0.91111	0.91111	0.97037	0.97037

Tabell 8 visar en jämförelse mellan studiens rerank-metod och en generell rerankermodell baserad på cross-encoder BGE-reranker-large).

Resultatet visar att studiens rerank-metod generellt uppnår högre värden över samtliga utvärderingsmått jämfört med cross-encodern. Detta indikerar en förbättrad förmåga att rangordna relevanta segment i de översta positionerna, samt en bättre generell förmåga att identifiera relevanta segment.

Cross-encodern uppvisar generellt lägre prestanda än querybaserade text utan rerank för flera av de utvärderade måtten. En marginell förbättring observeras dock för primär först, där värdet ökar från 0.54815 till 0.55556. För övriga mått ses en generell försämring jämfört med den strukturerade texten utan rerank.

5.1.3 Retrievalresultat totalt

I detta avsnitt visas en sammanfattad tabell av retrievalresultatet.

Tabell 9: *Sammanfattning av retrievalresultat för samtliga experimentella konfigurationer.*

Konfiguration	MRR	Primär först	Recall@1	Recall@3	Recall@5	Recall@10	Recall@20
Träningsset							
Strukturerad text	0.72580	0.40000	0.59259	0.82963	0.90370	0.96296	0.97778
Strukturerad text + rerank	0.85365	0.72593	0.78518	0.91111	0.91111	0.97037	0.97037
Querybaserad text	0.81312	0.54815	0.74074	0.87407	0.91111	0.96296	0.97778
Querybaserad text + rerank	0.86096	0.74815	0.80000	0.91111	0.91111	0.97037	0.97037
Querybaserad text + cross-encoder	0.78291	0.55556	0.71852	0.82963	0.87407	0.91852	0.92593
Testset							
Strukturerad text	0.74981	0.33191	0.64255	0.84823	0.89362	0.94043	0.94752
Strukturerad text + rerank	0.87728	0.68652	0.82128	0.92340	0.93617	0.96312	0.96738
Querybaserad text	0.80791	0.41560	0.73475	0.87376	0.91064	0.95035	0.95745
Querybaserad text + rerank	0.87281	0.68794	0.81135	0.92482	0.93617	0.96596	0.97021
Testset (med Doclingfel)							
Querybaserad text	0.78967	0.40136	0.71837	0.85442	0.88980	0.92653	0.93333
Querybaserad text + rerank	0.84354	0.65850	0.77959	0.89660	0.91293	0.94150	0.94558

Tabell 9 sammanfattar samtliga genomförda retrievalexperiment och deras respektive resultat över både tränings- och test-goldset. Tabellen möjliggör en direkt jämförelse mellan samtliga experiment och dess resultat över de olika konfigurationerna av goldset, rerank och struktur på tabellsegments text.

Sammanfattat uppvisar resultaten inom retrievalsteget att rerank genomgående bidrar till en förbättrad prestanda. Detta är särskilt tydligt på utvärderingsmåtten MRR och recall@1-3. Strukturen på tabellsegmentens retrievaltext (quertbaserad- eller strukturerad text) visar i flera fall på ytterligare förbättringar i samtliga mått för konfigurationer utan rerank, men dessa skillnader minskar för högre recall@k värden. Vidare visar resultaten en relativ stabil överensstämmelse mellan tränings- och test-goldset, vilket indikerar en stabilitet i metoden.

5.2 Extraktion

Resultaten för extraktionen är uppdelade i tre delmoment. I första delmomentet presenteras resultaten från tabellextraktionen. I andra delmomentet redovisas resultaten för QA-modellen och dess prestanda vid textextraktion. Slutligen sammanfattas de två tidigare delmomenten i en gemensam översikt av extraktionsresultaten.

I samtliga utvärderingar av extraktion observeras att Exact Match (EM) och F1-score är identiska, med undantag för träningen av QA-modellen där en avvikelse förekommer. I de nedanstående tabellerna visas därför endast ett av måtten.

Det är viktigt att påpeka att benämningarna ”querybaserad text” och ”rerank” utgår från retrievalsteget och avser variationer i den indata som extraktionen baseras på, snarare än förändringar i själva extraktionsmodellen. Extraktionsresultaten är därmed beroende av de segment som hämtas i föregående steg, vilket innebär att skillnader i retrieval kan påverka utfallet i extraktionen.

Samtliga resultatdata, består endast av de segment som goldset innehåller, vilket betyder att segment som extraheras som inte finns i goldset inte utvärderas.

5.2.1 Tabellextraktion

I detta delmoment utvärderas extraktionen enbart på tabellsegment. De extraherade värdena kommer från retrievalsteget, och extraktionsförmågan utvärderas för respektive konfiguration.

Resultatet delas upp i tre kategorier baserat på tabelltyp: tabeller som endast innehåller tydliga år, tabeller med blandad uppsättning eller år i spann samt tabeller utan

årsinformation. Utöver detta redovisas även ett totalt resultat för respektive konfiguration.

Tabell 10: *Extraktionsresultat - tränings-goldset.*

Konfiguration	Bara år	Blandade/År i spann	Inga år	Totalt
Querybaserad text	0.9805	0.7308	0.4000	0.8984
Querybaserad text + rerank	0.9849	0.7083	0.3600	0.8952

Tabell 10 visar resultatet för tränings-goldset med querybaserad text, med och utan rerank.

Resultaten visar att skillnaderna mellan konfigurationerna är relativt små sett till det totala måttet, som representerar den övergripande extraktionsprestandan. För querybaserad text observeras en marginell skillnad mellan konfigurationerna med och utan rerank, särskilt för kategorierna blandade/år i spann, 0.7308 jämfört med 0.7083, samt inga år, 0.4000 jämfört med 0.3600.

För de olika tabelltyperna kan en tydlig variation observeras. Tabeller med endast år visar en högre extraktionsförmåga än både blandade/år i spann och inga år. Den lägsta extraktionsförmåga kan ses i kategorin inga år, oavsett konfiguration. Resultaten tyder på att tabellens struktur har stor betydelse för modellens extraktionsförmåga.

Tabell 11: *Extraktionsresultat - test-goldset.*

Konfiguration	Bara år	Blandade/år i spann	Inga år	Totalt
Strukturerad	0.9833	0.9242	0.8358	0.9569
Querybaserad text	0.9835	0.9263	0.8447	0.9587
Strukturerad + rerank	0.9896	0.9492	0.8122	0.9652
Querybaserad text + rerank	0.9938	0.9489	0.8122	0.9682
Querybaserad text (Med Doclingfel)	0.9309	0.9027	0.8010	0.9121
Querybaserad text + rerank (Med Doclingfel)	0.9414	0.9105	0.7525	0.9171

Tabell 11 visar resultatet för test-goldset med fyra olika konfigurationer: med och utan rerank samt querybaserad text och strukturerad text.

Resultaten visar en generellt hög prestanda över samtliga konfigurationer, med en total extraktionsförmåga mellan 0.9575 och 0.9685 för goldset utan Doclingfel. Samtliga konfigurationer visar en likvärdig utvärdering för varje tabelltyp, där tabeller med bara år uppvisar den högsta extraktionsförmågan, följt av blandade/år i spann och sist tabelltypen inga år. Skillnaden mellan de olika konfigurationerna är små, men en svag förbättring kan observeras för konfigurationer med rerank, särskilt i kombination med querybaserad text. Resultaten tyder på att tabellens struktur har stor betydelse för modellens extraktionsförmåga även för testsetet. Vidare visar resultaten på att rerankingens påverkan på inkommande segment bidrar till en variation i extraktionsresulatet.

För test-goldset med Doclingfel uppvisas en tydlig minskning i samtliga kategorier, främst i tabelltypen inga år. Detta indikerar att fel i dokumentkonvertering påverkar extraktionsförmågan längre ner i pipelinen.

Tabell 12: *Extraktionsresultat för tabellextraktion i pipelinen.*

Konfiguration	Totalt
Träningsset	
Strukturerad text	0.8984
Strukturerad text + rerank	0.8952
Testset	
Baseline	0.9569
Strukturerad text	0.9587
Baseline + rerank	0.9652
Strukturerad text + rerank	0.9682
Testset (med Doclingfel)	
Querybaserad text	0.9121
Querybaserad text + rerank	0.9171

Tabell 12 sammanfattar resultaten för tabellextraktion över tränings- och test-goldset samt för fall med Doclingfel.

Resultaten visar övergripande stabila resultat över samtliga konfigurationer och goldset. De totala värdena för samtliga konfigurationer visar en liknande trend med små skillnader i extraktionsförmåga. På tränings-goldsetet observeras en marginell minskning vid användning av rerank, medan test-goldsetet visar en svag förbättring. Detta är särskilt påtagligt för konfigurationen med querybaserad text och rerank.

För test-goldsetet med Doclingfel observeras en tydlig försämring jämfört med motsvarande fall utan fel i konverteringen, 0.9685 mot 0.9171, vilket indikerar att kvaliteten i dokumentkonverteringen har en direkt påverkan på extraktionsprestandan.

5.2.2 Textextraktion

I detta delmoment utvärderas extraktionen enbart på textsegment. De extraherade värdena kommer likt tabellsegmenten från retrievalsteget, och extraktionsförmågan utvärderas för respektive konfiguration.

Först presenteras resultatet för QA-modellen XLM-RoBERTa-base-squad2. Därefter presenteras en sammanfattande tabell över textextraktionen för respektive konfiguration.

Tabell 13: *Dataset för träning av QA-modellen.*

Dataset	Antal datapunkter
Träningsset	4644
Valideringsset	643
Testset	580

Tabell 13 visar storleken samt fördelning av de datapunkter som använts vid träning, validering och testning av QA-modellen. Träningssetet utgör majoritet av den totala datan, medan validerings- och testset består av en mindre del.

Tabell 14: *Träningsresultat för QA-modellen XLM-RoBERTa-base-squad2 över tre epoker.*

Epoch	Eval Loss	Exact Match	F1
1	0.18668	0.9751	0.9759
2	0.13730	0.9782	0.9790
3	0.13622	0.9736	0.9743

Tabell 14 visar träningsresultaten för QA-modellen (XLM-RoBERTa-base-squad2) över tre epoker. Resultaten visar en stabil utveckling av eval loss med en mindre förbättring i början av träningen, följt av en stabilisering av både Exact Match (EM) och F1-score.

Tabell 15: *Resultat för modellen på modellens testsetet*

	Exact Match	F1
Testset	98.62%	98.62%

Tabell 15 visar modellens resultat på osedd data (modellens testset) med EM- och F1-värde på 0.9862. Resultatet indikerar att modellen har en god förmåga att identifiera korrekta svarsspann i relevanta kontexter. Resultatet visar att modellen presterar väl på data som följer liknande struktur som träningsdatan.

Detta resultat bör dock inte tolkas som ett direkt mått på pipelinens extraktionsprestanda utan som ett mått på modellen extraktionsförmåga under kontrollerade förhållanden.

Tabell 16: *Sammanfattning av extraktionsresultat för QA-modellen i pipelinen över tränings- och test-goldset.*

Konfiguration	Totalt
Träningsset	
Querybaserad text	0.9474
Querybaserad text + rerank	0.9500
Testset	
Strukturerad text	0.6590
Querybaserad text	0.6561
Strukturerad text + rerank	0.6703
Querybaserad text + rerank	0.6577
Testset (med Doclingfel)	
Querybaserad text	0.6422
Querybaserad text + rerank	0.6417

Tabell 16 visar ett sammanfattat resultat för QA-modellens prestation på tränings- och test-goldsetet, samt för fall med Doclingfel. Resultatet presenteras för de fyra tidigare konfigurationerna.

Resultatet visar att modellens prestation på tränings-goldsetet uppnår en hög och stabil prestanda med små marginella skillnader mellan konfigurationerna, 0.9474 samt 0.9500. Test-goldsetet uppvisas generellt lägre resultat för samtliga konfigurationer, där högsta värdet visar 0.6703. Detta tyder på att modellen har begränsad generaliseringsförmåga på osedd data.

För test-goldsetet med Doclingfel kan en marginell skillnad observeras, vilken kan indikera att dokumentkonverteringen har en större påverkan på resultatet än skillnader i modellkonfigurationer.

Tabell 17: *Sammanfattning av tabellextraktion, textextraktion och total extraktion.*

Konfiguration	Tabell	Text	Totalt
Träningsset			
Querybaserad text	0.8984	0.9474	0.9018
Querybaserad text + rerank	0.8952	0.9500	0.8993
Testset			
Strukturerad text	0.9575	0.6590	0.8965
Querybaserad text	0.9594	0.6561	0.8980
Strukturerad text + rerank	0.9556	0.6703	0.8992
Querybaserad text + rerank	0.9685	0.6577	0.8974
Testset (med Doclingfel)			
Querybaserad text	0.9121	0.6422	0.8594
Querybaserad text + rerank	0.9171	0.6417	0.8559

Tabell 17 visar en sammanfattad bild över resultaten från de olika extraktionsformerna: tabellextraktion, textextraktion samt den totala extraktionsprestandan över tränings- och test-goldset. Den inkluderar även fall med Doclingfel.

Resultaten visar att tabellextraktionen generellt har en hög och stabil prestanda över samtliga konfigurationer. Textextraktionen visar skilda värden, främst mellan tränings- och test-goldset, med de högsta redovisade resultatvärdena för respektive set är 0.9500 och 0.6703. Den totala extraktionsprestandan ligger däremellan, vilket återspeglar skillnaderna mellan de två tidigare delmomenten.

Skillnaderna mellan konfigurationerna är generellt små inom respektive delmoment. På tränings-goldset observeras små variationer mellan strukturerad text och reranking, medan test-goldset uppvisar en svag förbättrad prestanda vid användning av rerank i vissa konfigurationer.

För test-goldsetet med Doclingfel kan en tydlig minskning i tabellextraktionens prestanda observeras, vilket även kan utläsas av en lägre total extraktionsprestanda. Detta indikerar att kvaliteten i dokumentkonverteringen har märkbar påverkan på kvaliteten i de segment som skickas vidare till extraktionen.

5.3 Triangulering

Resultatet för trianguleringen bygger på resultaten från både retrieval- och extraktionsstegen och sammanfattas i Tabell 18. Resultatet delas upp i fyra kategorier baserat på svarets konfidens: Konsensus, klar vinnare, mjuk vinnare och konflikt. Utöver detta redovisas även EM för samtliga konfigurationer.

Resultaten påverkas därmed indirekt av både retrievalkvaliteten och extraktionsprestandan, eftersom trianguleringen baseras på de svar som genererats i tidigare steg.

Tabell 18: *Trianguleringsresultat för olika pipelinekonfigurationer på testdata.*

Konfiguration	EM	Konsensus	Klar vinnare	Mjuk vinnare	Konflikt
Träningsset					
Querybaserad text	0.7985	0.0000	0.6642	0.1049	0.2313
Querybaserad text + rerank	0.7612	0.0000	0.6791	0.1049	0.2090
Testset					
Strukturerad text	0.6985	0.0000	0.6147	0.1456	0.2397
Querybaserad text	0.6868	0.0000	0.6265	0.1500	0.2235
Strukturerad text + rerank	0.7147	0.0000	0.6483	0.1500	0.2015
Querybaserad text + rerank	0.7015	0.0000	0.6544	0.1471	0.1985
Testset (Med Doclingfel)					
Querybaserad text	0.6485	0.0000	0.6137	0.1520	0.2343
Querybaserad text + rerank	0.6625	0.0000	0.6402	0.1506	0.2092

Tabell 18 visar resultatet för trianguleringssteget över olika konfigurationer på träning- och test-goldset, samt med och utan Doclingfel.

Resultaten visar att ingen av konfigurationerna uppnår en tydlig konsensus mellan källorna. Detta indikerar att svaren i stor utsträckning skiljer sig åt mellan de underliggande källorna.

Bland kategorierna dominerar *klar vinnare* och *konflikt*, där andelen klar vinnare varierar mellan 0.6147 och 0.6544, medan konfliktandelen varierar mellan 0.1985 och 0.2397 för konfigurationerna i test-goldsetet utan Doclingfel. Modellen finner i majoriteten av fallen ett tydligt mest sannolikt svar, medan cirka 20–25 % av fallen innehåller konflikter mellan olika källor. Detta indikerar att årsredovisningarna ofta innehåller variationer mellan olika representationsformer av samma KPI.

För konfigurationerna utan Doclingfel observeras att användning av rerank generellt ger en högre andel klar vinnare och en lägre andel konflikt jämfört med konfigurationerna utan rerank. Detta indikerar en viss förbättring i hur väl relevanta källor identifieras och hur väl de överensstämmer, även om skillnaderna mellan konfigurationerna är relativt små.

För experimenten med Doclingfel observeras en övergripande försämring i EM jämfört med motsvarande konfigurationer utan Doclingfel. Detta tyder på att fel i dokumentkonverteringen påverkar stabiliteten i hela pipelinen och därmed även trianguleringsresultaten.

6 Diskussion

I detta avsnitt diskuteras pipelinen i sin helhet utifrån observerat resultat.

6.1 Docling och adaptiv segmentering

En central komponent i pipelinen är den adaptiva segmenteringsmodellen som behandlar den konverterade dokumentrepresentationen från Docling. Kvaliteten på den resulterade dokumentstrukturen från Docling är avgörande för efterföljande segmentering. Detta gör att fel som introduceras i Doclingkonverteringen får exempelvis felaktiga tabellstrukturer eller förlorade tabellrelationer och därför en direkt inverkan på segmentering, vilken leder till påföljder i efterföljande steg.

Den radvisa segmenteringen för tabeller resulterar i att segmenten ofta blir semantiskt homogena, vilket underlättar matchning mot specifika KPI:er i retrievalsteget.

Samtidigt kan radsegmenteringen riskera att relationer mellan rader och kolumner går förlorade. I mer komplexa tabeller, där värden, rubriker och aggregerade rader som ”Totalt” eller ”Summa” förekommer, blir detta särskilt tydligt, då ett bredare sammanhang ofta behövs för att kunna tolkas korrekt.

Ett alternativt tillvägagångssätt hade kunnat vara att använda sig av större segment för att bevara tabellens interna relationer och struktur. Detta skulle kunna ge förbättringsmöjligheter för kontextförståelse, men samtidigt skapas en ökning i mängden brus som segmenten innehåller. Det i sin tur kan försvåra processen att isolera exakta KPI-matchningar i retrievalsteget. Det här visar en tydlig avvägning mellan semantisk precision och strukturell kontext, därför prioriterar studiens metod semantisk precision för att nå mer träffsäkra resultat.

Mer specifika problem med Docling och den efterföljande segmenteringen har även observerats. Docling har visat sig fungera väl för standardiserade och väldefinierade tabeller, men variationer i tabellstruktur som flera rubriksnivåer, otydliga hierarkier, svåra tabellstrukturer och rader som saknar ”radetikett” skapar svårigheter för Docling att konvertera årsredovisningen. Detta har lett till att Doclingfel har uppstått och visat sig ha en påverkan på resultatet.

Metadata från konverteringssteget bevaras inte alltid konsekvent, vilket är ett återkommande problem som behövt tas i beaktande under segmenteringen. Särskilt påverkat är kategoriseringen av nivå. Därför har gränser behövts sättas under segmenteringen för att försöka representera korrekt nivåbyte (koncern och moderbolag). Om denna strukturella information saknas eller är inkorrekt påverkar det både segmenteringen och efterföljande steg i pipelinen, då det resulterar i att ofullständig kontext används för klassificering och tolkning av källor.

Även om dokumentkonverteringen i kombination med segmentering har visat sig innebära vissa strukturella utmaningar, visar resultaten att pipelinen i sin helhet ändå presterar stabilt.

Systemet uppvisar genomgående hög retrievalprestanda, medan exktraktionsresultat är mer varierande mellan tabell- och textsegment. Det indikerar att den valda segmenteringsmetoden i hög grad lyckas bevara relevant information till efterföljande steg i pipelinen.

Samtidigt tyder resultaten på att dessa fel har en påverkan på resultatet, vilket begränsar systemets maximala prestanda särskilt för tabellstrukturer.

6.2 Retrieval

Tre olika modelluppsättningar utvärderades för retrievalsteget och resultatet visade att både BGE-M3-modellerna generellt visar bättre prestanda än E5 på samtliga utvärderingsmått, med undantag för måttet primär först där E5 visade bättre resultat. BGE-M3 med enbart dense visade bättre resultat på mått som utvärderar segmentets placering i de övre positionerna i rankinglistan (MRR, primär först och recall@1) medan BGE-M3 med både sparse- och densevektor presterar bättre på resterande mått. Eftersom efterföljande steg i pipelinen, speciellt trianguleringen, bygger vidare på de segment som hämtats av retrievalsteget är det speciellt viktigt att de relevanta segment återfinns inom top-20. Den exakta placeringen i rangordningen blir därmed av begränsad betydelse, vilket grundar valet av BGE-M3 med både sparse- och dense för studiens retrievalsteg.

Resultaten för retrievalsteget visar överlag en stabil prestanda för studiens olika konfigurationer. När det kommer till respektive utvärderingsmått är variansen som högst för utvärderingsmåtten MRR, primär först, recall@1 och recall@3, vilket indikerar att

förändring i konfigurationer främst påverkar den initiala rangordningen av de översta resultaten, utan att ha en stor påverkan på den övergripande recallförmågan.

För övriga utvärderingsmått är skillnaderna genomgående relativt små. Dessa skillnader tyder på att retrievalmodellen uppvisar höga resultat för recall@10-20, vilket indikerar att relevanta segment i hög grad återfinns inom de hämtade segmenten. Den höga prestandan begränsar utrymmet för förbättringar för dessa utvärderingsmått.

Resultaten visar att retrievalsteget generellt presterar bra för både tränings- och testdata. I majoriteten av fallen rankas relevanta segment högt bland retrievalkandidaterna. Detta kan ses i MRR-värden där värdena ligger mellan 0.74 och 0.87 för test-goldsetet, vilket är lovande för nästkommande extraktionssteg. Även värden för recall@k visar att relevanta segment vanligtvis hamnar högt upp i rankingen. För utvärderingsmåttet recall@20 (relevanta källor) kan det utläsas att majoriteten av relevanta segment återfinns bland top-20 resultat i både tränings- och test-goldset. Utvärderingsmåtten precision@20 (korrekta källor) visar även att cirka 0.44 till 0.47 av top-20 kandidater består av relevanta segment med korrekt värde för test-goldsetet. Denna siffra ligger mellan 0.64 och 0.67 för tränings-goldsetet. Eftersom precision@20 (korrekta källor) mäter andelen relevanta segment som även innehåller korrekt värde i top-20 kan denna skillnad inte enbart direkt förklaras av skillnaden i goldsetens konstruktion. Eftersom en öknings i recall@20 (relevanta källor) även observeras skulle dessa resultat kunna förklaras genom att testsetet innehåller större variation i hur finanisell information presenteras, vilket leder till en ökning av recall@20 (relevanta källor) men att mängden källor som innehåller korrekt värde är mindre än tränings-goldsetet vilken ger en minsking i precision@20 (korrekta källor).

I resultaten kan en märkbar skillnad mellan recall@1 och recall@20 observeras, vilket innebär att retrievalsteget ofta hittar rätt segment, men inte alltid placerar det rätta segmentet högst upp. Detta motiverar användningen av top-k retrieval och efterföljande reranking- och trianguleringssteg i pipelinen.

6.2.1 Träningsset och Testset

Resultaten visar att samtliga utvärderingsmått är genomgående likartade för både tränings- och testsetet. Särskilt recall@10 och recall@20 uppvisar liknande resultat med små variationer. Detta indikerar att retrievalmodellen har en relativ god generaliseringsförmåga när det kommer till både sedd och osedd data.

Det ska dock noteras här att testsetet innehåller en större mängd potentiella källor per fråga jämfört med träningssetet. Det innebär att testsetet är mindre strikt definierad och mer inkluderande än för träningssetet, vilket har en påverkan på jämförbarheten mellan dessa. Då det finns flera annoterade relevanta källor blir recall svårare att tolka och jämföra direkt med träningssetet.

Denna skillnad i goldset innebär att resultatet för testsetet kan stabiliseras eller marginellt förbättras när flera relevanta segment inkluderas, då sannolikheten att lokalisera ett relevant segment ökar. Detta kan ses på de studiespecifika måtten (primärkälla @20, precision@20 (korrekta källor) och recall@20 (relevanta källor)) där primär källa@20 och precision@20 (korrekta källor) uppvisar högre resultat, samt ett försämrat resultat för recall@20 (relevanta källor) för träningsetet jämfört med testsetet. Ett förbättrat resultat på recall@20 (relevanta källor) är förväntat då en utökning av goldset ger fler relevanta källor för modellen att identifiera.

Trots dessa skillnader i goldsetkonstruktion uppvisar modellen jämförbara resultat mellan tränings- och testsetet, vilket indikerar att retrievalmodellen har en relativ stabil prestanda mellan de olika goldseten.

När det kommer till de högst rankade segmenten visar resultaten på en liknande trend för både tränings- och testset. Detta betyder att ett relevant segment i stor utsträckning hamnar högt i kandidatlistan. Eftersom samma mönster uppvisas för båda goldseten tyder detta på att retrievalmodellen bibehåller en relativt god förmåga att identifiera och rangordna relevanta segment.

6.2.2 Rerank och utan rerank

En viktig aspekt i analysen är effekten av reranking på retrievalresultatet. Detta jämförs mellan olika konfigurationer, med och utan heuristisk rerank, samt ett experiment med en cross-encoder rerank-modell.

Resultaten visar att rerank genomgående har en positiv effekt på resultatet, vilket framförallt är tydligt i de högst rankade segmenten. För övriga mått, recall@5-20 observeras endast marginella variationer i resultatet.

Detta indikerar att rerank främst påverkar precision i toppkandidaterna, utan att ha en stor inverkan på den övergripande retrievalförmågan.

För utvärderingsmåtten primärkälla@20, precision@20 (korrekta källor) och recall@20 (relevanta källor) kan det utläsas att rerank främst har en positiv påverkan på primärkällans position i retrievalsteget, vilket gäller för de båda dataseten, men särskilt tydligt för testsetet.

Detta beror troligtvis på att rerank inte introducerar nya kandidater, utan omordnar de redan hämtade kandidaterna från hybridsökningen. Effekten av rerank är därför beroende av hur väl den initiala hybridsökningen lyckas identifiera relevanta segment.

För cross-encoder modellen på querybaserad text, visar resultaten generellt en sämre prestation än hybridmodellen för querybaserad text utan rerank. Denna variation blir mest påtaglig för högre recall@k-nivåer. Detta indikerar att cross-encoder modellen i detta fall inte bidrar till någon förbättring utan snarare har en negativ inverkan på resultatet i stort. En rimlig förklaring till detta skulle kunna vara att cross-encodern inte är tränad på specifikt finansiella dokument och därför har en begränsad förmåga att avgöra relevans i tabellintensiva och domänspecifika segment. Detta är särskilt påtagligt i denna studie då många rerank-modeller tränats på löpande text, och inte strukturintensiva tabellsegment som denna studie använder sig av. En annan möjlig förklaring skulle kunna vara att då hybridmodellen i grunden visar relativt starka resultat, kan cross-encodern börja rangordna kandidater som redan ligger relativt rätt, vilket i stället kan få en negativ effekt på resultatet.

6.2.3 Skillnad mellan querybaserad text och strukturerad text

Resultaten visar generellt små skillnader mellan querybaserad- och strukturerad text, vilket tyder på att båda representationstyperna fångar relevanta segment i liknande grad. Resultatet skiljer sig främst på MRR, primär först och recall@1, vilket indikerar att formatet på tabelltexten har en liknande påverkan på retrievalresultatet som rerank. Utifrån detta kan det tolkas som att querybaserad text gör KPI:t mer explicit när det följer samma struktur som den ställda sökfrågan. Detta kan vara en anledning till att dessa segment rankas högre initialt från hybridsökningen än den strukturerade texten. Denna effekt har dock, precis som för rerankingen, ingen påverkan på retrievalkvaliteten på top-20 kandidater.

För måtten primär källa@20, precision@20 (korrekta källor) och recall@20 (relevanta källor) finns det marginella variationer mellan querybaserad- och strukturerad text. Detta är främst tydligt för primär källa@20, vilket skiljer sig med ca 2%. Denna

skillnad är dock så pass liten att ingen slutsats kan dras om huruvida dessa texttyper har någon påverkan på dessa mått.

Då recall@10-20 redan ligger mellan 0.95 och 0.98 från retrievalsteget nära en ”takeffekt”, vilket innebär att förmågan att identifiera rätt segment närmar sig sin maximala nivå och små förbättringar har minimal påverkan på resultatet. Detta gör att skillnad i textembeddings mellan querybaserad och strukturerad text, visar marginell skillnad i recall@10-20. Detta styrker argumentet att den querybaserade texten bidrar mer som ett rankingmått än förbättring av retrievalkvaliteten.

6.2.4 Med och utan Doclingfel

Ett extra test-goldset inkluderades för att utvärdera påverkan av felkällor vid dokumentkonverteringen med Docling. Resultaten indikerar att samtliga utvärderingsmått försämras när dessa Doclingfel förekommer. Även om variationen är liten påverkas retrievalsteget av Doclingfel. Detta är särskilt viktigt för recall@10-20, då detta kan leda till att vissa segment går förlorade eller identifieringsgraden blir sämre.

Resultaten visar att Doclingfel leder till en konsekvent försämring i retrievalprestandan. Effekten är dock relativt begränsad, vilken indikerar att retrievalmodellen i många fall fortfarande kan lokalisera relevanta segment trots brister i dokumentkonverteringen. Detta beror på att Doclingfel ofta förekommer på enskilda segment, vilket gör att andra relevanta segment fortfarande kan lokaliseras. Detta skapar då en begränsad effekt på resultatet.

Samtidigt behöver en begränsad påverkan på retrievalsteget inte innebära att efterföljande steg i pipelinen blir opåverkade. Doclingfel har inte enbart påverkan på retrievaltexten utan även det innehåll som skickas vidare till extraktionssteget. Denna påverkan diskuteras vidare i extraktionsavsnittet.

6.3 Extraktion

Utifrån Tabell 17 kan en tydlig skillnad observeras mellan resultaten för extraktion från tabellsegment och extraktion från textsegment. Extraktion från tabeller uppvisar genomgående högre prestanda än extraktion från löpande text. Detta kan ses som lovande då de flesta finansiella KPI:er från årsredovisningar ofta förekommer i tabeller, främst huvudtabeller som balansräkning, resultaträkning eller kassaflödesanalys.

En möjlig förklaring till denna skillnad är att tabeller ofta är mer strukturerade än löpande text. Finansiella KPI:er presenteras ofta tillsammans med tydliga radetiketter, kolumnrubriker och årtal, vilket gör det möjligt för den regelbaserade extraktionsmetoden att identifiera och koppla samman rätt KPI med rätt värde på ett förutbestämt sätt. Risken för tolkningsfel blir därmed lägre än vid extraktion från text.

Textsegment är däremot mer varierande när det kommer till språk och struktur. Samma KPI kan uttryckas på flera olika sätt, i olika formuleringar samt med olika synonymer. Även flera värden kan förekomma i samma mening och komplexiteten kan variera. Detta gör att höga krav behöver ställas på den transformbaserade QA-modellen, som då måste förstå den semantiska kontexten och identifiera rätt svarsspann i texten där flera rimliga kandidater kan förekomma.

Resultaten tyder på att den valda hybridstrategin är väl anpassad för uppgiften, då respektive extraktionsmetod är lämpad för den information den hanterar. Systemet visar sig prestera bäst på delar av dokumentet där primärinformation vanligtvis hittas.

Det ska dock nämnas att resultaten från extraktionen utgår från de segment där goldset har facit, vilket betyder att den inte helt återspeglar den totala extraktionsför-

mågan för alla segment. Det kan dock utläsas från Tabell 4 och 6 att andelen segment som innehar facit ligger mellan cirka 0.65 och 0.8, vilket tyder på att en majoritet av de hämtade segmenten innehåller facit, vilket då styrker extraktionsresultatets validitet.

6.3.1 Tabellextraktion

Extraktionsresultaten för tabeller visar att metodens prestation beror mycket på hur tabellen är formaterad. För en tabell med tydliga år som kolumnrubriker extraheras rätt värde i de allra flesta fall, vilket kan utläsas i resultaten för utvärderingsmåttet bara år. Även för tabeller med blandade/ år i spann visade sig extraktionsmetoden i de flesta fall fungera väl. En minskning i extraktionsförmåga kan dock noteras för kategorin inga år. Utifrån dessa resultat kan en trend tydas, där extraktionsresultatet påverkas märkbart beroende på hur årtalen är presenterade.

Resultatet visar en liknande extraktionsförmåga över samtliga utvärderingspunkter och konfigurationer. Detta är väntat då samtliga konfigurationer inte har någon direkt påverkan på extraktionsmodellen. Dessa variationer i resultat uppstår därmed på grund av små variationer i retrievalstegets resultat. Då resultaten visar ett liknande resultat från extraktionen, indikerar detta att det i stor utsträckning identifieras överlappande segment i det tidigare steget. Detta gör att indata för extraktionen blir liknande mellan konfigurationerna, vilket begränsar skildringar i resultat.

Utifrån resultaten kan det konstateras att Exact Match-värdet är något högre för testsetet än för träningssetet. Detta är särskilt noterbart för kategorin ”inga år” där en ökning från 0.36–0.40 för träningssetet till 0.81–0.84 för testsetet kan noteras. Detta är något oväntat resultat, då modeller generellt tenderar att prestera bättre på data de har tränats på. En möjlig förklaring till detta resultat är att goldset för testsetet kompletterades med flera sekundärkällor. Eftersom ett extraherat värde kan förekomma på flera platser i en årsredovisning innebär fler annoterade sekundärkällor att systemet har fler möjligheter att få en korrekt träff vid utvärderingen. Detta kan i sin tur bidra till högre EM-värden.

En annan möjlig förklaring är uppbyggnaden av testsetet. Testsetet består av 15 årsredovisningar fördelade över fem olika bolag, vilket innebär att samma bolag hanteras flera gånger. Flera årsredovisningar inom samma bolag kan ha en tendens att dela liknande struktur, terminologi och presentationssätt. Om årsredovisningarna har många strukturella likheter kan retrieval- och extraktionssteget gynnas av att samma typ av KPI återfinns i liknande sektioner och tabellstrukturer. Detta kan göra det enklare för systemet att identifiera relevanta källor och extrahera korrekta värden, vilket kan bidra till att testsetet visar på högre resultat än träningssetet. Resultatet skulle även kunna förklaras av att olika typer av tabeller inkluderas i respektive dataset, vilket innebär att svårighetsgraden för tabellextraktionen inte nödvändigtvis är jämförbar mellan tränings- och testdataseten.

Utifrån studiens experiment är det därmed svårt att avgöra vilken av dessa faktorer som påverkar resultaten mest, men det indikerar på att resultatet inte är helt jämförbara mellan tränings- och testset.

Resultatet från de olika konfigurationerna visar att Doclingfel skapar en tydlig effekt på extraktionsförmågan. För konfigurationen querybaserad text kan det utläsas en minskning i EM för tabellextraktion från cirka 0.96 till cirka 0.91, vid introduktion av Doclingfel. Detta kan förklaras genom att extraktionssteget är starkt beroende av strukturen och innehållet på de inkommande segmenten. När Doclingfel introduceras i form av felaktig tabellstruktur, sammanfogade celler eller förlorad relation mellan KPI, årtal och värde, försämras kvaliteten på den information som segmenteringen bygger

på. Detta leder till att extraktionen får svårare att extrahera rätt information från datan.

Konsekvensen av Doclingfel blir därför inte bara en fråga om modellens prestanda, utan även en konsekvens av att kvaliteten på indata till extraktionssteget försämras. Detta gör att även om retrievalsteget lyckas i sin process att identifiera relevanta segment, kan segmentet innehålla felaktig information.

Sammanfattat tyder detta på att resultatet från dokumentkonvertering skapar en viktig felkälla i pipelinen. Detta då den påverkar informationskvaliteten redan innan extraktionssteget används.

6.3.2 Textextraktion

För textextraktion visar resultatet för träningssetet sig vara märkbart bättre än för testsetet. Detta tyder på att den finjusterade XLM-RoBERTa-modellen har lättare att identifiera korrekta svarsspann i textsegment som liknar de exempel som modellen tränats på. Till skillnad från tabellextraktion, där informationen ofta följer en relativt konsekvent struktur, består löpande text ofta av större variation i språk, formuleringar och kontext. Samma KPI kan uttryckas på flera olika sätt och i olika meningskonstellationer, vilket gör generalisering till nya dokument mer utmanande.

Resultaten tyder på att QA-modellen är något känslig för variationer i formuleringar och dokumentstruktur, vilket är särskilt tydligt i testsetet där en minskad extraktionsförmåga kan noteras. En bidragande faktor kan även vara den begränsade mängden träningsdata, vilket innebär att modellen saknat tillräckligt med exempel att lära sig varierande sätt att uttrycka finansiell information på. Detta kan leda till att modellen presterar väl på textsegment som liknar träningsdatan, men sämre på osedda och varierande formuleringar i testdatan.

Resultaten tyder även på att extraktion ur text är mer utmanande än extraktion ur tabeller. Medan tabeller ofta presenterar information i ett strukturerat format där relationen mellan KPI och värde är tydlig, kräver textextraktion både semantisk förståelse och korrekt identifiering av KPI och svarsspann i en mer komplex kontext. Detta gör att skillnader mellan tränings- och testresultat blir mer framträdande för textsegment än för tabellsegment.

En anledning till detta är att träningsdatan som modellen tränats på hade förberetts för träning, vilket då bestod av enklare meningar, främst med ett enda KPI. Detta skulle då kunna påverka resultatet för testdatan då texten ofta kan sakna tydlighet, flera språkliga variationer och uppställningar av KPI med en eller flera KPI i samma mening. Detta kan då påverka extraktionsförmågan för modellen på osedd data, vilket skulle kunna beskriva resultatet.

Resultaten visar att textextraktionen är betydligt mindre påverkad av Doclingfel, där enbart en marginell försämring observeras. Detta tyder på att textsegment i mindre utsträckning påverkas av strukturella fel i dokumentkonverteringen, då denna typ av segment inte är beroende av explicit tabellstruktur.

Dessa resultat bör dock inte enbart tolkas utifrån modellens egna prestation utan kan även ses från ett pipeline perspektiv. Detta betyder att modellen även är beroende av tidigare steg i pipelinen, vilket betyder att brister i segmenteringen och retrievaln kan ha en påverkan på resultatet.

Resultaten bör därför inte tolkas som ett direkt mått på QA-modellens extraktionsförmåga, utan snarare som ett mått på pipelinens totala prestation för retrieval- och extraktionsstegen.

6.4 Triangulering

Trianguleringssteget i denna studie fungerar som ett beslutslager där extraherade värden aggregeras och vägs samman utifrån uppskattad tillförlitlighet. Detta lägger då grunden för ett beslut av det slutgiltiga värdet. Till skillnad från föregående steg, där fokus ligger på enskilda segment, syftar trianguleringen till att hantera situationer där likvärdiga källor innehåller varierande numeriska värden.

EM ligger generellt mellan cirka 0.65 och 0.80 beroende på goldset, vilket tyder på en relativ stabil prestanda över samtliga konfigurationer. Det kan dock noteras en tydlig skillnad mellan tränings- och testset, där EM minskas från cirka 0.76–0.80 på träningssetet till cirka 0.69–0.71 på testsetet utan Doclingfel. Detta tyder på att även trianguleringssteget är beroende av tidigare stegs prestanda i pipelinen. Eftersom trianguleringssteget inte har någon egen förmåga att påverkan tidigare stegs prestation är den enbart beroende av kandidatgenereringen och resultatet från pipelinens tidigare steg. Detta reducerar möjligheten för trianguleringen att identifiera korrekt svar när felaktiva kandidater introduceras eller relevanta segment missas. När Doclingfel introduceras minskas EM ytterligare till cirka 0.64–0.66. Detta styrker argumentet att tidigare stegs prestation påverkar trianguleringens förmåga att identifiera det mest sannolika värdet bland flera oberoende evidenskällor.

Utifrån resultaten kan observeras att fördelningen av beslutsstatus är relativt likartad mellan konfigurationerna. Andelen klar vinnare är den mest framträdande, som utgör majoriteten av fallen, cirka 0.61–0.68, medan mjuk vinnare och konflikt representerar resterande del. Detta tyder på att systemet i de allra flesta fall lyckas särskilja ett dominerande kandidatvärde från övriga. Samtidigt förekommer fall med motstridiga kandidatvärden, vilket indikerar att viss osäkerhet kvarstår.

En viktig detalj i resultatet är avsaknaden av andelar konsensusfall för samtliga konfigurationer. Detta bör dock inte tolkas som en svaghet i trianguleringen utan beror sannolikt på hur pipelinens retrievalsteg är utformad. Eftersom kandidatgenerering från retrievalsteget alltid returnerar ett fast antal segment per datapunkt, inkluderas även i många fall fler källor än vad som är relevanta. Detta gör att en fullständig överenskommelse mellan extraherade värden sällan uppstår, vilket förklarar resultatet.

Resultatet visar även att skillnader mellan konfigurationer är varierande men begränsade. Detta kan bero på att, likt extraktionen, är trianguleringen beroende av föregående steg i pipelinen. Det tyder på att de olika konfigurationerna inte har någon direkt påverkan på trianguleringsmodellen, utan variationerna kan förklaras av skildringar i resultat från retrieval- och extraktionssteget. Eftersom dessa steg i stor utsträckning genererar överlappande segment mellan konfigurationerna blir även trianguleringens indata likartad, vilket begränsar skillnaderna i slutresultatet.

Sammanfattat visar trianguleringen att genom en sammanvägning av flera evidenskällor kan tillförlitligheten förbättras, men metoden är beroende av tidigare steg i pipelinen och kan inte kompensera för fel som uppstår i dessa steg.

6.5 Begränsningar

Datamängden för denna studie var något begränsad, vilket kan påverka generaliserbarheten hos systemet. Som träningsdata användes 29 svenska årsredovisningar och testsetet bestod av 15 svenska årsredovisningar. Även om resultatet för retrievalsteget är lovande är det svårt att fullt ut bedöma systemets prestation på större och mer varierande dokumentmängder med olika språk, layoutstrukturer och branschvarierande årsredovisningar.

En annan begränsning handlar om att synonymlistor och mappningar mellan KPI och tabelltyp är manuellt konstruerade. Detta introducerar ett beroende där dessa kopplingar kräver bred domänkunskap. Detta kan resultera i falskt positiva och falskt negativa klassificeringar.

En begränsning i studiens utvärdering är att två olika strukturer på goldset använts i denna studie. Tränings-goldsetet bestod av ett striktare facit där enbart segment som innehöll rätt värde togs med. Test-goldsetet utvidgades och fler relevanta segment inkluderades. Detta innebär att vissa retrievalmått, särskilt recallbaserade mått, inte är fullt jämförbara mellan dataseten eftersom fler segment som anses vara relevanta inkluderas i test-goldsetet.

En begränsning som är påtaglig för studien är respektive stegs beroende av korrekt dokumentkonvertering och segmentering. Resultaten visar att samtliga steg i pipelinen förbättras när felaktiga segment från Docling exkluderas. Detta indikerar att delar av systemets fel orsakas av brister i dokumentkonverteringen. Årsredovisningar innehåller ofta en komplex och variationsrik layout med tabeller, noter, textsegment, punktlistor etc. Detta gör att felaktig dokumentkonvertering kan leda till felaktig segmentering och informationsförlust, vilket främst är påtagligt på tabeller. Även tabeller kan ha många olika format och former, vilket försvårar segmenteringen och kan påverka resultatet i studien.

En embeddingbaserad retrieval möjliggör semantisk sökning men kan fortfarande ha begränsad förståelse för numeriska relationer och specifika finansiella uttryck. Detta kan leda till att semantiskt liknande men irrelevanta dokumentsegment rankas högt.

En annan begränsning i denna studie är att systemets pipeline är starkt beroende av retrievalsteget eftersom efterföljande extraktion endast genomförs på de hämtade kandidaterna. Om relevanta dokumentsegment inte identifieras i retrievaln kan efterföljande steg inte kompensera för detta, vilket gör retrievalsteget till en kritisk flaskhals i systemet.

Retrievalsteget använder sig av ett fast antal hämtade segment (top-20). Detta kan introducera irrelevant brus och ge modellen mer kandidatmaterial som kan påverka både resultatet i retrievalsteget men även pipelinens senare steg, då fler källor kan skapa svårare beslutsvariabler.

Vidare bör recall@k tolkas med försiktighet för testsetet, då den endast utvärderar hur väl retrievalmodellen identifierar ett relevant segment och inte huruvida detta segment innehåller det korrekta värdet. Då höga recall@20 värden för detta set inte nödvändigtvis betyder hög slutlig korrekthet bör därför retrievalresultaten tolkas tillsammans med resterande delar av pipelinen (extraktion och triangulering), vilka bättre speglar systemets förmåga att producera korrekta slutgiltiga svar.

För tabeller baseras extraktionssteget i huvudsak på regelbaserade metoder och heuristisk matchning. Detta gör systemet känsligt för variationer i tabellstruktur, KPI-benämningar och dokumentformat.

En radbaserad retrieval används för tabeller i denna studie, vilket kan innebära informationsförluster. Årsredovisningar kan innehålla komplexa tabellstrukturer där relationer mellan rader, kolumner och hierarkiska rubriker kan förloras i den konverterade dokumentrepresentationen, speciellt vid radbaserad retrieval.

En begränsning med trianguleringen är att den bygger en heuristisk poängsättning, vilket betyder att den använder fasta och inte inlärda vikter. Detta kan leda till låg generaliserbarhet till nya domäner i framtiden.

Slutligen returnerar systemet ett enskilt numeriskt värde per fråga. Även om triangulerings steget beräknar tillförlitligheten heuristiskt (konsensus eller konflikt), modelleras osäkerheten inte som ett statistiskt konfidensintervall eller en sannolikhetsfördelning

över möjliga svar, utan som en regelbaserad uppskattning av tillförlitlighet.

Trots flera begränsningar visar resultaten att retrievalbaserad KPI-extraktion från årsredovisningar är möjlig.

7 Slutsats

Syftet med denna studie var att undersöka hur väl en RAG-baserad pipeline kan identifiera, lokalisera och extrahera finansiella KPI:er från årsredovisningar. För att uppnå detta utvecklades en pipeline som kombinerar retrieval-, extraktions- och trianguleringsmetoder för att hantera både strukturerad och ostrukturerad information i årsredovisningar.

Resultatet visar att en hybridbaserad embeddingmodell generellt presterar bättre i retrievaluppgifter. BGE-M3 visar de bästa resultaten när det kommer till MRR- och recall-mått jämfört med Multilingual-E5-large. Detta tyder på att en kombination av sparse- och dense-representationer lämpar sig väl i domäner som finansiella dokument där både semantiska likheter och lexikala egenskaper har stor betydelse.

Vidare visar resultaten att rerank har en påverkan på segmentens ordning, men påvisar mindre effekt på det totala antalet identifierade segment från hybridsökningen. Resultaten visar även att typen av retrievaltext för tabellsegment har en påverkan på systemets prestanda. Den querybaserade representationen uppvisar generellt bättre resultat och mer stabil prestanda jämfört med strukturerad text i konfigurationer utan rerank. Samtidigt visar resultatet att skillnader mellan texttyperna minskar när rerank appliceras, vilket indikerar att rerank bidrar till ett jämnare retrievalresultat mellan de olika konfigurationerna.

Vidare visar resultaten att en generell cross-encoder-baserad rerank-modell BGE-reranker-large presterar sämre än den heuristiska rerank-modellen som används i denna studie. Detta indikerar att domänspecifik heuristik är mer effektiv än generella modeller i finansiella dokument, där domänspecifik kunskap har en stor betydelse.

Resultaten för extraktionen i pipelinen är generellt goda, där tabellextraktion visar bättre prestation än textextraktion med avseende på extraktionsförmåga. Resultaten för textextraktionen med en transformerbaserad QA-modell visar att metoden kan användas för att extrahera svar från givna textsegment. Modellen lyckades identifiera relevanta KPI:er i många fall, men resultaten visar samtidigt att fel fortfarande förekommer. Detta indikerar att QA-modeller kan extrahera finansiell information från kontexter där KPI:er och värden kan presenteras i varierande formuleringar och sammanhang, då semantisk förståelse är nödvändig.

Vidare visar resultaten för tabellextraktionen en generell hög prestanda för samtliga konfigurationer, med stabila värden för både EM och F1-score. Detta indikerar att finansiella tabeller i stor utsträckning kan extraheras korrekt när rätt segment väl har identifierats. Resultatet visar att tabellformatet har en påverkan på extraktionsförmågan. För tabeller med tydligt angivna år uppvisas en nästintill perfekt extraktionsförmåga. Tabeller utan årtal visar däremot sämre resultat. Detta beror på att tabeller i årsredovisningar är varierande och komplexa, där en mer kontextuell förståelse över sambanden mellan rader och kolumner kan behövas. Vidare visar resultaten att fel vid dokumentkonverteringen har en negativ påverkan på systemet prestanda. Detta visar att kvaliteten i den initiala dokumentkonverteringen är en betydande faktor för den slutliga extraktionskvalitet.

Sammantaget visar resultaten att tabellextraktionen i en RAG-baserad pipeline är relativ robust, där den största begränsningen grundar sig i retrieval- och preprocessingsteget, snarare än i själva extraktionsmodellen.

Resultaten för trianguleringen visar att systemet i relativt hög grad lyckas presentera ett korrekt slutvärde. Det resultaten även visar är att skillnader mellan källor förekommer, vilket tyder på att finansiell information ofta uttrycks på olika sätt i årsredovisningar. Detta motiverar implementeringen av en trianguleringsbaserad valideringsmetod.

Resultaten visar även att en källa identifieras som klar vinnare i majoriteten av fallen, samtidigt förekommer flera fall av konflikt mellan källor. Detta tyder på att samma KPI, med varierande representationer återfinns i olika delar av årsredovisningen. Även i trianguleringen observeras att Doclingfel har en betydande roll för resultaten, vilket även styrker att den initiala dokumentkonverteringen är viktig genom hela pipelinen. Sammantaget visar resultaten att triangulering är ett effektivt verktyg för att sammanväga flera informationskällor där informationen är inkonsekvent snarare än att verifiera identiska svar.

Resultaten från denna studie förväntas bidra till kunskap om hur hybrida retrievalmetoder och evidensbaserad triangulering kan förbättra tillförlitligheten vid informationsextraktion från finansiella dokument och har potentiella tillämpningar inom kreditriskbedömning och anomalidetektion inom banksektorn.

8 Framtida arbete

Framtida arbeten kan riktas mot flera delar av den föreslagna pipelinen. Då studien hanterar hela pipelinen från PDF till slutgiltigt svar har en del förbättringsmöjligheter identifierats. En central förbättring är att hantera begränsningar i dokumentkonverteringen, med särskilt fokus på tabellstrukturering. Förbättrad robusthet i detta steg skulle sannolikt minska flera fel i pipelinens efterföljande steg.

Vidare skulle en mer robust segmenteringsmodell kunna implementeras. Den nuvarande segmenteringsmodellen behandlar direkt den konverterade dokumentrepresentationen och dess utdata är starkt beroende av dess kvalitet. Detta medför att fel i exempelvis Docling får en direkt påverkan på segmenteringssteget, vilket därmed påverkar resten av pipelinen. Den nuvarande segmenteringsmodellen bygger på en radbaserad segmentering av tabeller, vilket innebär en viss strukturell informationsförlust, särskilt i fall där relationen mellan rader och kolumner inte är tydlig. Framtida arbeten skulle därför kunna fokusera på att utveckla en mer strukturmedveten segmenteringsmodell, som bevarar tabellers interna relationer och hierarkier mer effektivt.

Ett annat framtida arbete skulle kunna utgå ifrån att förbättra retrievalsteget genom att utveckla och finjustera rerank-modeller på domänspecifik data. De resultat som observerades med en generell reranker indikerar att modeller utan domänanpassning inte nödvändigtvis leder till förbättrad prestanda i denna kontext, vilket motiverar behovet av mer specialiserade lösningar. En annan metod hade kunnat vara att testa att utveckla en ”Learning-to-Rank (LTR-modell)”, vilket är en modell som tränas på annoterad data för att optimera rangordningen av kandidater baserat på relevanssignaler. Detta skulle kunna möjliggöra en mer datadriven viktning av relevanta kandidatsegment än den nuvarande modellen som utgår från heuristiska rankingstrategier. En alternativ metod skulle kunna vara att välja större tabellsegment. Detta skulle då bidra med mer kontextuell information, särskilt i mer komplexa tabeller. Denna lösning riskerar dock att försämra precision i retrievalsteget och göra det svårare att isolera relevanta värden.

Ett annat framtida arbete skulle kunna gå ut på att förbättra trianguleringssteget genom mer sofistikerade beslutsmodeller, exempelvis probabilistisk aggregering eller inlärda viktningar. Detta skulle kunna minska känsligheten för motstridiga kandidatvärden och ytterligare stabilisera slutresultatet.

Något som skulle stärka generaliserbarheten och göra systemet mer robust är att vidga både tränings- och test-goldsetet med fler och varierande årsredovisningar och KPI:er. Ett framtida arbete kan även vara att undersöka prestandan för årsredovisningar skrivna på andra språk än svenska och utvidga till fler typer av bolag, exempelvis fastighetsbolag.

Litteraturförteckning

[1] Suleiman Abbadi and Sharif M. Abu Karsh. Methods of evaluating credit risk used by commercial banks in palestine. *International Research Journal of Finance and Economics*, 2013.

[2] Jumana Alsubhi, Mohammad Alahmadi, Ahmed E Alhusayni E, and Ibrahim Aldailami. Optimizing rag pipelines for arabic: A systematic analysis of core components. 2025.

[3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern information retrieval. 1999.

[4] David Balsiger, Hans-Rudolf Dimmler, Samuel Egger-Horstmann, and Thomas Hanne. Assessing large language models used for extracting tbale information from annual financial reports. *Computers*, 2014.

[5] Anita Bans-Akutey and Benjamin Makimilua Tiimub. Triangulation in research. 2021.

[6] Joel Barnard. What is embedding? n.d. Tillgänglig online: `https://www.ibm.com/think/topics/embedding` (2026-01-31).

[7] Olga Bogachek, Antonio De Vito, Paul Demeré, and Francesco Grossetti. Using narrative disclosures to predict tax outcomes. *Review of Accounting Studies*, 2025.

[8] Douglas Burdick, Marina Danilevsky, Alexandre V Evfimieevski, Yannis Katsis, and Nancy Wang. Table extraction and understanding for scientific and enterprise applications. *IBM Research*, 2020. Tillgänglig online: `https://www.vldb.org/pvldb/vol13/p3433-burdick.pdf` (2026-03-03).

[9] Nancy Carter, Denise Bryant-Lukosius, Alba DiCenso, Jennifer Blythe, and Alan J Neville. The use of triangulation in qualitative research. *Oncology Nursing Forum*, 2014.

[10] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. 2025.

[11] Georgios Chochlakis, Jackson Trager, Vedant Jhaveri, Nikhil Ravichandran, Alexandros Potamianos, and Shrikanth Narayanan. Semantic f1 scores: Fair evaluation under fuzzy class boundaries. 2025.

[12] Chanyeol Choi, Jihoon Kwon, Minsoo Ha, Alejandro Lopez-Lira, Minjae Kim, Chaewoon Kim, Yongjae Lee, Juneha Hwang, Jaeseon Ha, Suyeol Yun, and Jin Kim. Structuring the unstructured: A multi-agent system for extracting and querying financial kpis and guidance. 2025.

[13] Stefano Cirillo, Domenico Desiato, Giuseppe Polese, and Giandomenico Solimando. Benchmarking google embeddings 2 against open-source models for multilingual dense retrieval and rag systems. 2026.

[14] Gordon V Cormach, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. 2009.

[15] Phuong-Nam Dang, Kieu-Linh Nguyen, and Thanh-Hieu Pham. Viranker: A bge-m3 blockwise parallel transformer cross-encoder for vietnamese reranking. 2025.

[16] Paulo Roberto de Moura Júnior, Jean Lelong, and Annabelle Blangero. Adaptive chunking: Optimizing chunking-method selection for rag. 2026.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[18] Nikhil Dharmaram. Evaluate retrieval quality in rag pipelines: Mean reciprocal rank (mrr) and average precision (ap). 2025.

[19] Docling. Docling converts messy documents into structured data and simplifies downstream document and ai processing by detecting tables, formulas, reading order, ocr, and much more. 2025. Tillgänglig online: `https://www.docling.ai/` (2026-02-03).

[20] Andrea L. Eisfeldt and Gregor Schubert. Generative ai and finance. *Annual review of financial economics*, 2025.

[21] Mohammad Fawsi Shubita. Earnings and market ratio: additional evidence from jordania banks. *Banks and Bank Systems*, page 14, 2023.

[22] Christopher M. Frenz. Introduction to searching with regular expressions. 2008.

[23] Luyu Gau, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. 2021.

[24] Veknat N. Gudivada, Dhana L. Rao, and Amogh R. Gudivada. Information retrieval: Concepts, models, and systems. *Handbook of Statistics*, 2018.

[25] Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Lio, and Benyou Wang. An empirical study on information extraction using large language models. 2024.

[26] Yucheng Hu and Yuxing Lu. Rag and rau: A survey on retrieval-augmented language model in natural language processing. 2025.

[27] Daniel Jurafsky and James Martin H. Speech and language processing. 2026.

[28] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. 2020.

[29] Jay Kim. Unlocking the power of fuzzy matching in python: A practical guide. 2025.

[30] Hassane Lakhloufi. Recall@k evaluation. 2026. Tillgänglig online: url =https://metricgate.com/docs/recall-at-k-evaluation/ (2026-04-20).

[31] Baruch Lev. Industry averages as targets for financial ratios. *Journal of Accounting Research, Vol. 7, No 2*, page 290, 1969.

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. 2019.

[33] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieva*. Cambridge University Press, 2009.

[34] Christopher Manning D, Prabhakar Raghavan, and Schütze Hinrich. Introduction to information retrieval. 2008.

[35] Paulo Sérgio Medeiros dos Santos and Guilherme Horta Travassos. On the representation and aggregation of evidence in software engineering: a theory and believe based perspective. 2013.

[36] Milvus. The high-performance vector database built for scale. 2026. Tillgänglig online: `https://milvus.io/` (2026-02-10).

[37] Syed Farhan Mohsin, Syed Imran Jami, Shaukat Wasi, and Muhammad Shoaib Siddiqui. An automated information extraction system from the knowledge graph based annual financial reports. *Peer Computer Science*, 2023.

[38] Syed Farhan Mohsin, Syed Imran Jami, Shaukat Wasi, and Muhammad Shoaib Siddiqui. An automated information extraction system from the knowledge graph based annual financial reports. *PeerJ Computer Science*, 2024.

[39] Saeedeh Momtazi and Dietrich Klakow. A word clustering approach for language model-based sentence retrieval in question answering systems. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1911–1914. ACM, 2009.

[40] Madhurima Nath. Fuzzy matching algorithms. 2024.

[41] Tejul Pandit, Mahendru Sakshi, Meet Raval, and Upadhyay Dhvani. The evolution of reranking models in information retrieval: From heuristic methods to large language models. 2025.

[42] Mohamed Ibrahim Ragab, Ensaf Hussein Mohamed, and Walaa Medhat. Mulitlingual propaganda detection: Exploring transformer-based models mbert, xlm-roberta, and mt5. 2025.

[43] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+questions for machine comprehension of text. 2016.

[44] Zafaryab Rasool, Stefanus Kurniawan, Sherwin Balugo, Scott Barnett, Rajesh Vasa, Courtney Chesser, Benjamin M Hampstead, Sylvie Belleville, Kon Mouzakis, and Alex Bahar-Fuchs. Evaluating llms on document-based qa: Exact answer selection and numerical extraction using cogtale dataset. 2024.

[45] Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerruegel. Docparser: Hierarchical document structure parsing from rendering. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, page 1, Virtual Conference, 2021. AAAI Press.

[46] Jhon Rayo. A hybrid approach to information retrieval and answer generation for regulatory texts. 2025.

[47] Alexey O. Shigarov. Table understanding using a rule engine. 2015.

[48] Siddharth Pratap Singh. Cross-encoder models for enhanced search relevance: A multi-domain analysis of performance and applications. 2024.

[49] Jie Tang, Hong Mingcai, Duo Zhang, Bangyong Liang, and Juanzi Li. Information extraction: Methologies and applications. 2007.

[50] Evidently AI Team. Mean reciprocal rank (mrr) explained. 2025. Tillgänglig online: url =https://www.evidentlyai.com/ranking-metrics/mean-reciprocal-rank-mrr (2026-04-15).

[51] Vidar Tobrand. Financial kpi extraction using large language models. 2026.

[52] Mohamed Trabelsi, Zhiyu Chen, Shuo Zhang, Brian D. Davison, and Jeff Heflin. Strubert: Structure-aware bert for table search and matching. 2022.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Lilion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2023.

[54] Di Wang and Eric Nyberg. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 707–712, Beijing, China, 2015. Association for Computational Linguistics.

[55] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, and Charles Xie. Milvus: A purpose-built vector data managenemt system. 2021.

[56] Jiawei Wang, Kai Hu, Zhuoyao Zhong, Lei Sun, and Qiang Huo. Detect-order-construct: A tree construction based approach for hierarchical document structure analysis. 2024.

[57] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. 2024.

[58] Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. Evidence aggregation for answer re-ranking in open-domain question answering. *International conference on learning representation*, 2018.

[59] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications: A literature review. 2017.

[60] Abraham Itzhak Weinberg. Hybrid dense-sparse retrieval for high-recall information retrieval. 2026.

[61] Minji Wo and Amélie Marian. A framework for corroborating answers from multiple web sources. *Information Systems*, 2025.

[62] Yihengm Xu, Li Minghao, Cui Lei, Huang Shaohan, Wei Furu, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. 2020.

[63] Zhong Xu, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largets dataset ever for document layout analysis. 2019.

[64] Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1744–1753, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

[65] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjum Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024.

Appendix

A Hanteringsdetaljer av grupperade segment

Denna del beskriver hanteringen av grupperade segment för den adaptiva segmenteringen

A.1 Hantering av listor

Hanteringen av listor innebär att varje listpunkt transformeras till en rad i segmentets textrepresentation med prefixet ”-” och ett radslut. Nästkommande segment adderas successivt tills max-tokenlängden uppnås. Även här används samma metodik som för textsegment vid skapande av nya segment med överlappning och naturliga stopp.

A.2 Hantering av Nyckel-värde-par

Nyckel-värde-grupper behandlas som par av två element, ”nyckel” och ”värde”. Eftersom ordningen mellan värdet och nyckeln kan skifta (ibland förekommer värde före nyckel) används en metod (värdepoäng) för att identifiera vilket element som sannolikt är nyckel respektive numeriskt värde. Metoden värderar numeriska siffror högt och nedvärderar kontext med hög andel bokstäver. Därefter jämförs de två strängar som metoden ger som utdata, och sedan placeras de som antingen ”värde, nyckel” eller ”nyckel, värde” beroende på respektive värdepoäng. Därefter normaliseras paret till formatet ”Nyckel: Värde”, och eventuell förekomst av enheter extraheras och sparas med metadata. Precis som för textsegment avslutas dessa när token-längden maximeras. Därefter initieras ett nytt segment med ett överlapp på 60 tokens för att bibehålla kontexten. På så sätt kan listor och nyckel-värde-par bli sökbara semantiskt via embeddings, samtidigt som den ursprungliga strukturens ”nyckel-värde”-logik bibehålls i textrepresentationen och kompletteras med metadata.

B Milvus implementationsdetaljer

Detta avsnitt beskriver implementationsdetaljer kopplat till Milvus.

B.1 Milvus Schema

Följande kod visar Milvus-schema som användes för lagring av dokumentsegment. Dessutom visas de index som som skapades för semantisk och lexikal sökning.

```
milvus_schema.add_field(field_name="chunk_id",
      datatype=DataType.VARCHAR,
      is_primary=True,
      max_length=100,
      auto_id=False
      )

  milvus_schema.add_field(
      field_name="chunk_type",
      datatype=DataType.VARCHAR,
      max_length=100,
      enable_analyzer=False,
      enable_match=False
```

```
)
milvus_schema.add_field(
    field_name="text",
    datatype=DataType.VARCHAR,
    max_length=65535,
    enable_analyzer=False,
    enable_match=False
)

milvus_schema.add_field(
    field_name="extraction_text",
    datatype=DataType.VARCHAR,
    max_length=65535,
    enable_analyzer=False,
    enable_match=False
)


milvus_schema.add_field(
    field_name="embedding",
    datatype=DataType.FLOAT_VECTOR,
    dim=1024
    )

milvus_schema.add_field(
    field_name="sparse_embedding",
    datatype=DataType.SPARSE_FLOAT_VECTOR
)
milvus_schema.add_field(
    field_name="page",
    datatype=DataType.INT64,
    enable_analyzer=False,
    enable_match=False
)

milvus_schema.add_field(
    field_name="source_type",
    datatype=DataType.VARCHAR,
    max_length=100,
    enable_analyzer=False,
    enable_match=False
)

milvus_schema.add_field(
    field_name="section",
    datatype=DataType.VARCHAR,
    max_length=1000,
    enable_analyzer=True,
    enable_match=True
)
milvus_schema.add_field(
    field_name="header",
    datatype=DataType.VARCHAR,
    max_length=1000,
    enable_analyzer=True,
    enable_match=True
)
milvus_schema.add_field(
    field_name="level",
    datatype=DataType.VARCHAR,
    max_length=100,
    enable_analyzer=False,
```

```
    enable_match=False
)
milvus_schema.add_field(
    field_name="table_type",
    datatype=DataType.VARCHAR,
    max_length=200,
    enable_analyzer=False,
    enable_match=False
)
milvus_schema.add_field(
    field_name="unit",
    datatype=DataType.VARCHAR,
    max_length=100,
    enable_analyzer=False,
    enable_match=False
)
milvus_schema.add_field(
    field_name="table_kpi",
    datatype=DataType.VARCHAR,
    max_length=1000,
    enable_analyzer=False,
    enable_match=False
)
milvus_schema.add_field(
    field_name="table_id",
    datatype=DataType.VARCHAR,
    max_length=100,
    enable_analyzer=False,
    enable_match=False
)

milvus_schema.add_field(
    field_name="values",
    datatype=DataType.ARRAY,
    element_type=DataType.VARCHAR,
    max_capacity=20,
    max_length=1000
)
milvus_schema.add_field(
    field_name="columns",
    datatype=DataType.ARRAY,
    element_type=DataType.VARCHAR,
    max_capacity=20,
    max_length=1000
)
milvus_schema.add_field(
    field_name="years",
    datatype=DataType.ARRAY,
    element_type=DataType.VARCHAR,
    max_capacity=20,
    max_length=1000
)
milvus_schema.add_field(
    field_name="column_headers_raw",
    datatype=DataType.JSON,
    max_capacity=20,
    max_length=1000
)
milvus_schema.add_field(
    field_name="row_values_raw",
    datatype=DataType.JSON,
    max_capacity=20,
```

```
      max_length=1000
  )
```

Samt följande index:

```
index_params.add_index(
      field_name="embedding",
      index_type="AUTOINDEX",
      metric_type="IP",
      index_name="embedding"
  )

  index_params.add_index(
      field_name="sparse_embedding",
      index_type="SPARSE_INVERTED_INDEX",
      metric_type="IP",
      index_name="sparse_index"
  )
  index_params.add_index(
      field_name="header",
      index_type="INVERTED",
      index_name="header"
  )

  index_params.add_index(
      field_name="section",
      index_type="INVERTED",
      index_name="section"
  )
```

B.2 Indexeringsval för Milvus

Embeddingvektorn indexeras med en storlek representativt till modellens embedding-dimension. Indextyp i Milvus för embeddingvektorn sätts till AUTOINDEX, vilket tillåter Milvus att själv bestämma vilken ANN-sökning som är mest optimal.

Sparse index (sparse-vektor) sätts till SPARSE_INVERTED_INDEX. Detta tillåter Milvus att söka efter alla aktiva termer utifrån sökfrågan och gör en inverterad sökning mot segment som innehåller samma termer i årsredovisningen. Resultatet baseras på antalet term-matchningar.

C Implementationsdetaljer för hybridsökning

Denna sektion sektion går igenom implementationsdetaljer för hybridsökningen.

C.1 k-sweep

För att identifiera hur många segment som behöver extraheras från Milvus, görs en top-k sweep [30] på tränings-goldsetet. Eftersom två separata sökningar appliceras, en per segmenttyp, behöver två k-värden identifieras. Retrieval körs initialt med ett högt k-värde, varefter mindre top-k-nivåer simulerades offline genom trunkering av den rankade resultatlistan och utvärdering av retrievalkvaliteten för respektive k-cutoff.

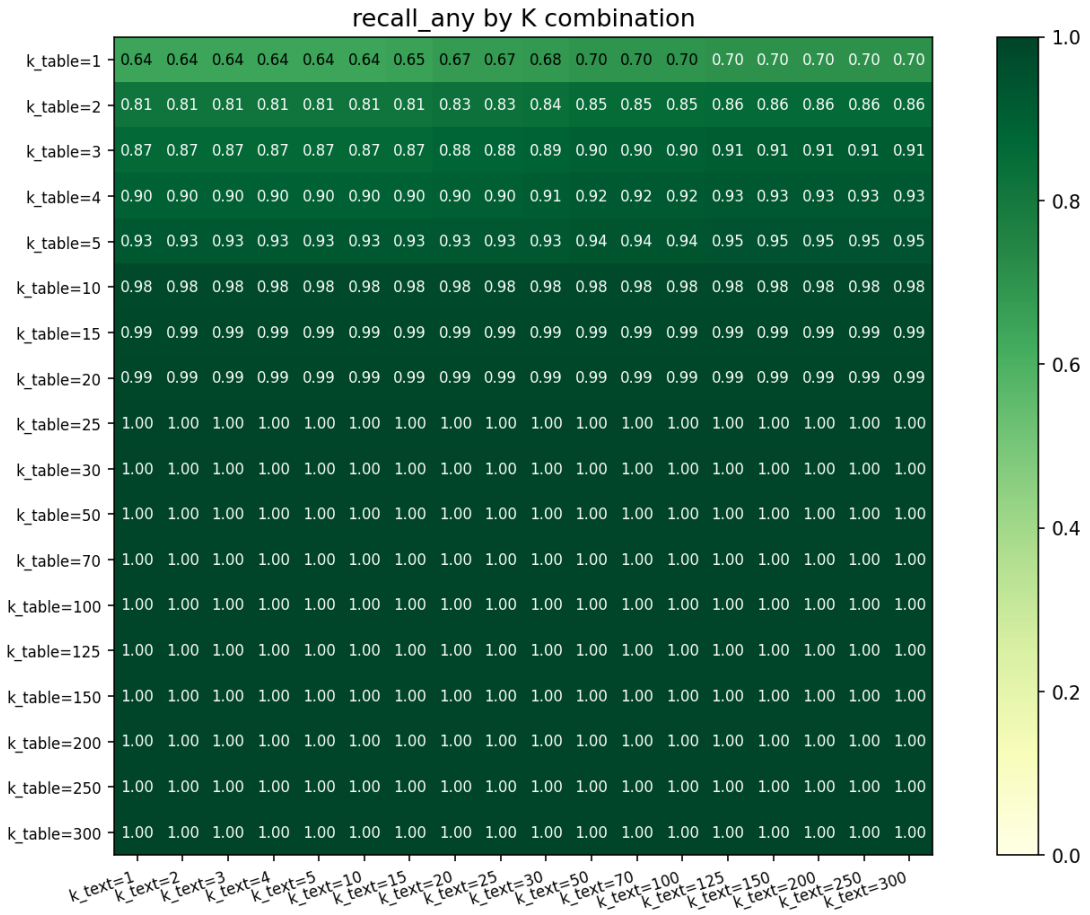
Tre olika scenarier analyseras:

1) Recall av minst ett relevant segment.

2) Recall då endast ett relevant segment finns att identifiera.

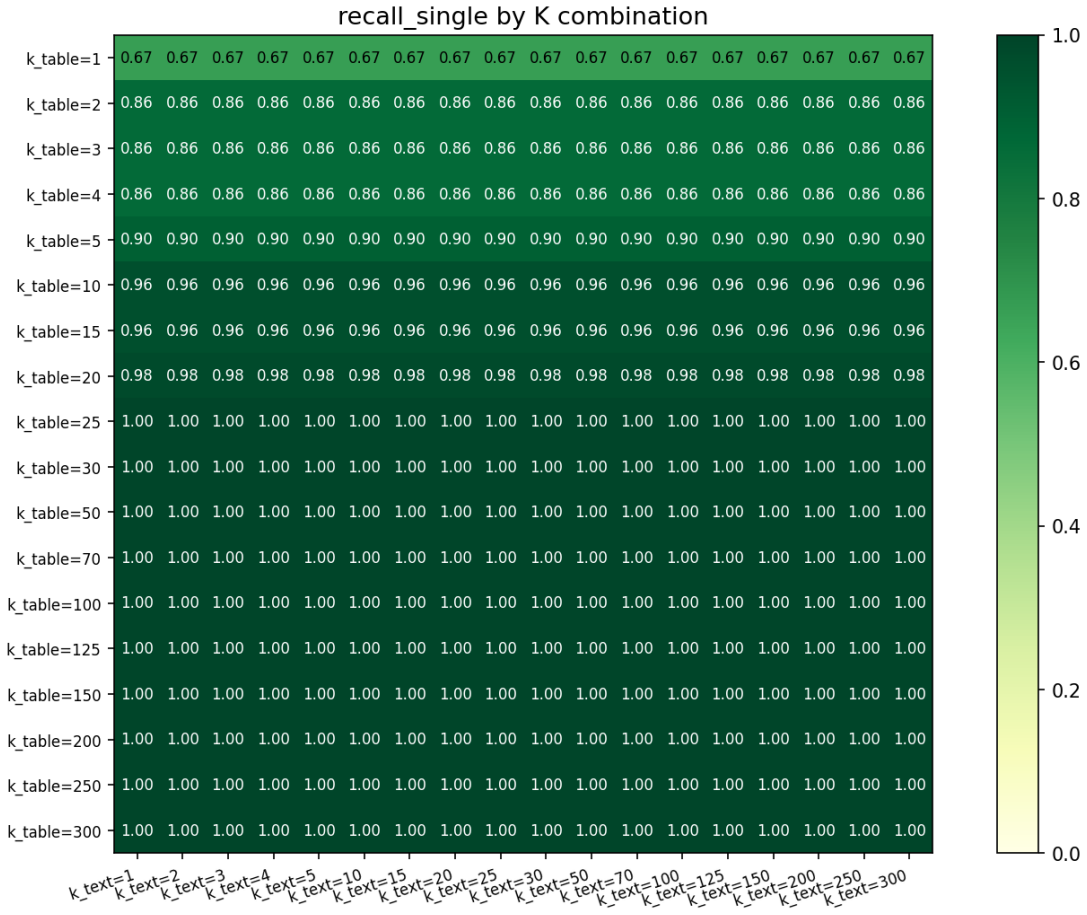
3) Recall då fler än två relevanta segment identifieras.

Resultaten av parameterstudien redovisas i Figur 6–8.



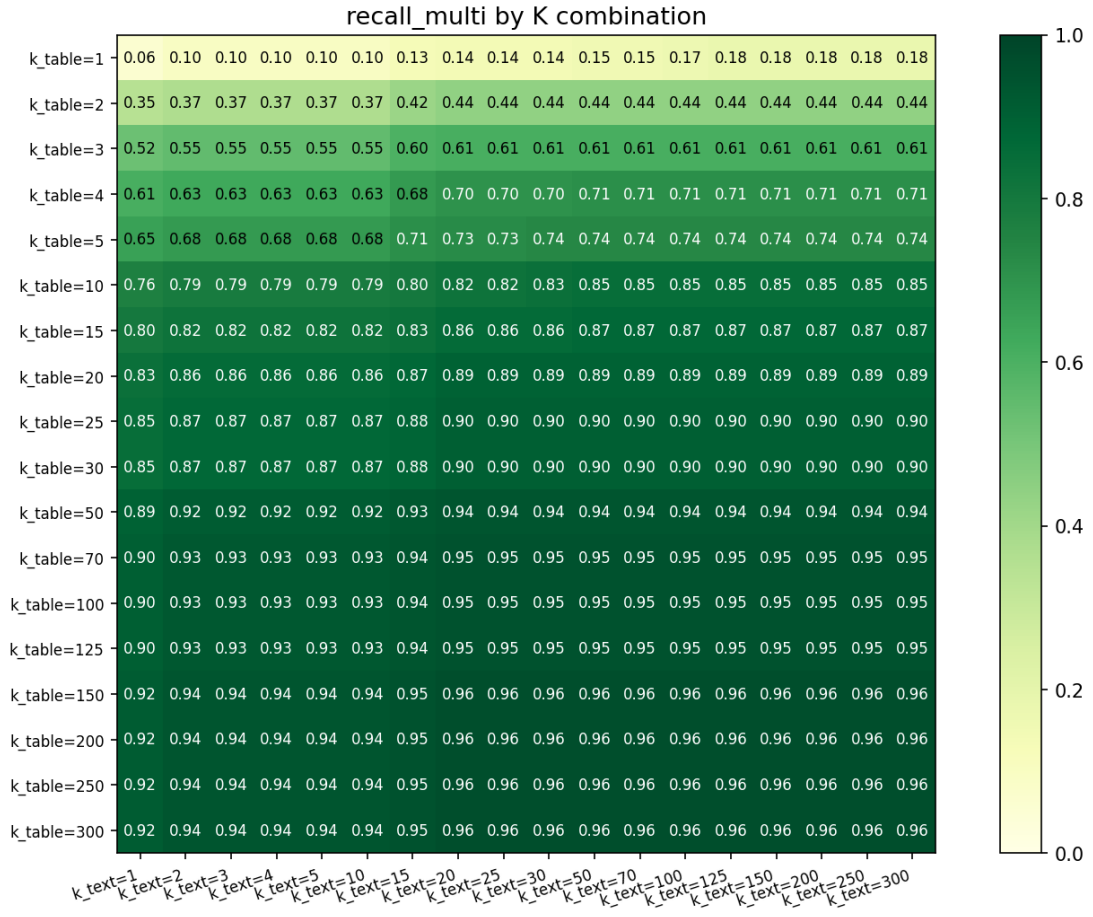
Figur 6: *Gräns för retrieval av tabellsegment och textsegment med hänseende på recall any.*

Figur 6 visar en heatmap över recall för minst ett relevant segment. Denna heatmap undersöker hur recall ändras då k-värdet ökar när något segment identifieras som även erhålls i goldset.



Figur 7: *Gräns for retrieval av tabellsegment och textsegment med hänseende på recall single.*

Figur 7 visar en heatmap över recall för fall då goldet endast innehåller den primära källan.



Figur 8: *Gräns for retrieval av tabellsegment och textsegment med hänseende på recall multi.*

Figur 8 visar en heatmap över recall för fall då två eller fler segment hämtades från retrievalsteget.

Eftersom den efterföljande triangulering bygger på en aggregering av flera evidenskällor, prioriteras tredje scenariot. De slutgiltiga k-värdena väljs därmed utifrån de värden där ytterligare ökningar ger begränsade förbättringar i recall.

D Implementationsdetaljer för reranking

Denna sektion går igenom implementationsdetaljer för reranking.

D.1 KPI-likhet för tabellsegment

För att bedöma hur väl ett tabellsegment överensstämmer med frågans KPI beräknas flera kompletterade likhetsmått. Detta beror på att dokumentkonverteringen kan introducera fel som stavfel, sammandragna ord och variationer i KPI-benämningar.

Följande likhetsmått används:
Token Sort Ratio används som grundmått för att undersöka tokenlikhet oberoende av ordning. Två exempel kan ses i Tabell 19.

Tabell 19: Exempel på applikation av token sort ratio

Fråga	Segment
Totala intäkter	Intäkter
Rörelsemarginal	Justerad rörelsemarginal

Token coverage används för att undvika höga poäng för segment med begränsad begreppsöverlappning. Detta beräknas enligt Ekvation 9.

$$\text{Coverage} = \frac{|A \cap B|}{\max(|A|, |B|)} \tag{9}$$

Tokenbaserad likhetspoäng är en kombination av dessa och skapar en slutgiltig likhetspoäng. Detta kan ses i Ekvation 10.

$$\text{Score}_{\text{token}} = \text{Score}_{\text{fuzzy}} * \text{Coverage} \tag{10}$$

Bokstavsbaserad likhet används som en fallback, vilket jämför strängar för varje tecken där mindre stavfel och variationer fångas. För att minska risken för falska positiva matchningar, viktas måttet med en faktor på 0.85.

Full merge similarity används som en annan fallback för fall då fel i dokumentkonverteringen slagit samman ord. Här jämförs då normaliserade KPI-strängar där mellanslag tagits bort med kontexten. Se exemplet i Tabell 20.

Tabell 20: Konverteringsexempel på borttagning av mellanslag

Innan konvertering	Efter konvertering
Summa tillgångar	summatillgångar
Totala intäkter	totalaintäkter

Slutgiltig KPI-likhet är det sammanslagna likhetsvärdet, vilket består av ett maxvärde av ovanstående likhetspoäng. Detta kan ses i Ekvation 11.

$$\text{Similarity} = max(\text{Score}_{\text{Token}}, \text{Score}_{\text{Karaktär}}, \text{Score}_{\text{FullMerge}}) \tag{11}$$

D.2 Synonymhantering

För varje KPI byggs en synonymlista upp, vilket används för att hantera variationer i finansiell terminologi. I Tabell 21 visas KPI med respektive synonymer.

Tabell 21: KPI och respektive synonymer

Begrepp	**Synonymer**
Försäljning av egna aktier	Köp/försäljning av aktier
Köp/försäljning av aktier	Försäljning av egna aktier
Amortering av långfristiga lån	Återbetalning av medel- och långfristiga lån
Återbetalning av medel- och långfristiga lån	Amortering av långfristiga lån
Övriga långfristiga skulder	–
Förändring i verkligt värde	Verkligt värdeförändringar
Verkligt värdeförändringar	Förändring i verkligt värde
Intäkter	Nettoomsättning, Försäljning, Omsättning, Nettoförsäljning, Totala intäkter
Nettoomsättning	Intäkter, Försäljning, Omsättning, Nettoförsäljning, Totala intäkter
Försäljning	Intäkter, Nettoomsättning, Omsättning, Nettoförsäljning, Totala intäkter
Omsättning	Intäkter, Nettoomsättning, Försäljning, Nettoförsäljning, Totala intäkter
Nettoförsäljning	Intäkter, Nettoomsättning, Försäljning, Omsättning, Totala intäkter
Betald ränta	–
Räntebärande skulder	Räntebärande nettoskuld
Långfristiga finansiella skulder	Långfristig upplåning, Finansiella skulder, Långfristiga låneskulder
Nettoskuld	Räntebärande skulder, Räntebärande nettoskuld
Räntebärande nettoskuld	Räntebärande skulder, Nettoskuld
Långfristiga räntebärande skulder	Långfristiga finansiella skulder, Långfristig upplåning
Långfristig upplåning	Långfristiga räntebärande skulder, Nettoskuld, Långfristiga finansiella skulder

Begrepp	Synonymer
Kortfristiga räntebärande skulder	Kortfristiga lån, Räntebärande kortfristiga skulder, Kortfristig upplåning
Kortfristiga finansiella skulder	Kortfristig upplåning, Finansiella skulder, Kortfristiga låneskulder
Kortfristiga lån	Kortfristiga räntebärande skulder, Kortfristiga finansiella skulder, Räntebärande kortfristiga skulder, Kortfristig upplåning
Räntebärande kortfristiga skulder	Kortfristiga räntebärande skulder, Kortfristiga finansiella skulder, Kortfristiga lån, Kortfristig upplåning
Kortfristig upplåning	Kortfristiga räntebärande skulder, Kortfristiga finansiella skulder, Kortfristiga lån, Räntebärande kortfristiga skulder
Goodwill	–
Minoritetsintresse	Innehav utan bestämmande inflytande, Icke kontrollerande intressen, Minoritet, varav innehav utan bestämmande inflytande
Betald utdelning	Utdelning till aktieägare, Aktieprogram netto, Utbetald utdelning
Kassaflöde från finansieringsverksamheten	Kassaflöde från finansieringsverksamhet, Nettokassaflöde från finansieringsaktiviteter, Finansieringsverksamhetens kassaflöde
Betald skatt	Betalda inkomstskatter, Skatter, Inkomstskatt
Totala tillgångar	Summa tillgångar, Balansomslutning

Begrepp	Synonymer
Totala intäkter	Summa intäkter, Nettoomsättning, Intäkter, Omsättning
Resultat efter finansnetto	Resultat före skatt
Årets kassaflöde	Nettoförändring av likvida medel innan valutaändringar, Reell förändring av likvida medel, Förändring likvida medel enligt bokslut
Eget kapital	Summa eget kapital, Totalt eget kapital
Soliditet	Soliditeten, Soliditet,%, EBIT-marginal

D.3 KPI-likhet för textsegment

Denna del går igenom hur KPI-likhet beräknas för textsegment.

Exakt träff beskriver fall då ett KPI eller tillhörande synonym hittas exakt i texten. Då returneras maximal likhetspoäng.

Fuzzy matching *partial ratio* används för att hitta delmatchningar mellan KPI och textsegment

Begreppstäckning används även för textsegment för att identifiera överlapp mellan KPI och texten. Detta visas i Ekvation 12.

$$\text{Coverage} = \frac{|A \cap B|}{|A|} \tag{12}$$

Kombinerad slutpoäng Den slutgiltiga likhetspoängen beräknas genom följande formel. Detta visas i Ekvation 13.

$$\text{Similarity} = 0.7 * \text{Partial}_{Ratio} + 0.3 * \text{Coverage} \tag{13}$$

D.4 Tröskelvärden

För att straffa svaga matchningar och belöna bra skapas vissa tröskelvärden. Segment med en likhetspoäng som överstiger tröskelvärden får en positiv viktning medan segment som kommer under får en negativ viktning. Dessa tröskelvärden redovisas i Tabell 22.

Tabell 22: Tröskelvärde för olika segmenttyper

Segmenttyp	Tröskel
Tabellsegment	60
Textsegment	65

För likhetspoäng ges även en extra positiv viktning för båda segmenttyperna med reducerande belöning beroende på respektive likhetspoäng. Detta görs för tröskelvärdena, 100, 80, 50, 30.

E Implementationsdetaljer för extraktion

Denna sektion går igenom implementationsdetaljer för extraktionssteget.

E.1 Val av värdekandidat för tabellsegment

För fall då flera värdekandidater finns bland kandidatgenereringen används en prioriteringslista, vilket kan ses i Tabell 23.

Tabell 23: Prioriteringslista för val av kandidater

Nyckelord	**Prioritet**
koncern	1
total	2
summa	3
utgående balans	4
årets slut	5
nettobelopp redovisade i	6
netto	7
redovisat värde	8
helår	9
verkligt värde	10
brutto	11
q4 / kv4	12
q3 / kv3	13
q2 / kv2	14
q1 / kv1	15
omräknat	16
förändring	17
gruppen	18
årets början	50

F Implementationsdetaljer för triangulering

Denna sektion går igenom implementationsdetaljer för trianguleringssteget.

F.1 Tabelltypsmappning

Trianguleringssteget använder sig av en tabelltyp-mapping för respektive KPI. Mappningen som gjorts kan ses i Tabell 24.

Tabell 24: KPI och tabelltypsmappning

KPI	**Rapporttyp**
Totala intäkter	resultaträkning
Intäkter	resultaträkning
Övriga intäkter	resultaträkning
Rörelseresultat	resultaträkning
Rörelsemarginal	resultaträkning
Finansiella intäkter	resultaträkning
Finansiella kostnader	resultaträkning
Räntekostnader	resultaträkning
Resultat efter finansnetto	resultaträkning
Resultat före skatt	resultaträkning
Skatt	resultaträkning
Betald skatt	kassaflödesanalys
Poster av engångskaraktär	resultaträkning
Varav poster av engångskaraktär	resultaträkning
Förändring i verkligt värde	resultaträkning
Goodwill	balansräkning
Totala tillgångar	balansräkning
Eget kapital	balansräkning
Eget kapital hänförligt till moderbolagets aktieägare	balansräkning
Minoritetsintresse	balansräkning
Obeskattade reserver	balansräkning
Soliditet	balansräkning
Nettokassa	balansräkning
Räntebärande skulder	balansräkning
Kortfristiga räntebärande skulder	balansräkning
Långfristiga räntebärande skulder	balansräkning
Långfristiga finansiella skulder	balansräkning
Kortfristiga finansiella skulder	balansräkning
Kortfristiga leasingskulder	balansräkning
Långfristiga leasingskulder	balansräkning
Övriga långfristiga skulder	balansräkning
Kortfristiga lån	balansräkning

KPI	Rapporttyp
Skulder till kreditinstitut	balansräkning
Kassa och bank	balansräkning
Kortfristiga placeringar	balansräkning
Likvida medel	balansräkning
Kassaflöde från den löpande verksamheten före förändring av rörelsekapital	kassaflödesanalys
Kassaflöde från förändring av rörelsekapital	kassaflödesanalys
Kassaflöde från investeringsverksamheten	kassaflödesanalys
Kassaflöde från investeringsaktiviteter	kassaflödesanalys
Kassaflöde från finansieringsverksamheten	kassaflödesanalys
Kassaflöde från finansieringsverksamhet	kassaflödesanalys
Kassaflöde från finansieringsverksamheten avseende främmande kapital	kassaflödesanalys
Förändring likvida medel enligt bokslut	kassaflödesanalys
Nettoförändring av likvida medel innan valutaändringar	kassaflödesanalys
Årets kassaflöde	kassaflödesanalys
Valutakursdifferenser	kassaflödesanalys
Upptagande av lån	kassaflödesanalys
Amortering av lån	kassaflödesanalys
Amortering av leasingskulder	kassaflödesanalys
Amortering av långfristiga lån	kassaflödesanalys
Betald utdelning	kassaflödesanalys
Återköp av egna aktier	kassaflödesanalys
Nyemission	kassaflödesanalys
Försäljning av egna aktier	kassaflödesanalys
Förvärv av innehav utan bestämmande inflytande	kassaflödesanalys