LLM-assisted trend scout

ALEXANDER SAE-CHO EK AND LUCAS MÅNSSON
BACHELOR´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Bachelor's Thesis

LLM-assisted trend scout

By

Alexander Sae-Cho Ek and Lucas Månsson

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Abstract

It is important for a large company like Bosch to be at the forefront of technological development, and a central part of this entails scouting for trends in the market. To achieve this, they search the Internet for news articles that could be of interest, providing them with vast amounts of information. A problem, however, is that they receive information about very many different areas in many different regions, which makes it difficult and time-consuming to sort out what may be interesting and relevant. This thesis explores how LLMs can be used to assist with finding relevant information on the internet.

The thesis contains two main parts: One explores how a pre-trained LLM can be optimized through different techniques to evaluate given information according to its relevance to Bosch. The other explores how this solution could be integrated into a web-based application. To test how an LLM could be optimized, 5 different LLM techniques (zero-shot prompt, few-shot prompt, multi-stage pipeline, RAG 1, and RAG 2) were found and created by doing literature research, and each LLM technique were tested on a test-set of articles that already were labeled on a scale from 0-3 by a Bosch employee were the score 0 meant not relevant and score 3 meant highly relevant. To test how this solution could be integrated into a web-based application, a prototype was made upon requirement from a Bosch employee and some user-tests were done by a Bosch employee.

The results of the different LLM techniques showed that multi-stage pipeline performed best in F1-score, Cohens kappa and Quadrant kappa in predicting the right relevant score to an article. But in another test where scores 2 and 3 were seen as positive relevance, and scores 0 and 1 were seen as negative relevance. Few-shot performed better than multi-stage in recall of articles with positive relevance, meaning it found most of the positively relevant articles. To minimize the risk of missing a relevant article in the prototype, a few-shot prompt would be the best LLM technique for the prototype.

Due to the limitation of training data in the RAG 1 technique, a reliable conclusion could not be drawn from the results for RAG 1, and since RAG 1 is similar to the few-shot technique, RAG 1 was used in the prototype although few-shot prompt performed better in the LLM-technique tests. But to make a better decision about which LLM technique to use in the prototype, more test data should be used to compare the different LLM techniques. The final prototype was tested by a Bosch employee according to the user-tests and the result showed that the prototype was easy to use and had a high number of relevant articles.

Keywords
Artificial Intelligence, Large Language Models, GPT-4o mini, OpenAI, Prompting, Retrieval Augmented Generation, Information Retrieval, Web development.

Sammanfattning

Det är viktigt för stora företag som Bosch att ligga i framkant när det gäller teknisk utveckling. En central del av detta innebär att bevaka trender på marknaden genom att söka på internet efter nyheter som kan vara av intresse, vilket ger dem stora mängder information. Ett problem är dock att de får information om väldigt många olika områden i många olika regioner, vilket gör det svårt och tidskrävande att sålla fram vad som kan vara intressant och relevant. Detta examensarbete undersöker hur en LLM kan användas för att filtrera relevanta och irrelevanta nyhetsartiklar.

Detta examensarbete består av två huvuddelar: Första delen utforskar hur en förtränad LLM kan optimeras med hjälp av olika tekniker för att utvärdera relevansen på en given information för bosch. Den andra delen utforskar hur denna lösning kan integreras till en webbaserad applikation. För att testa hur en LLM kan optimeras har 5 olika LLM-tekniker (zero-shot, few-shot, multi-stage pipline, RAG 1, och RAG 2) tagits fram med hjälp av litteraturundersökning. Varje LLM-teknik testades på ett testset av artiklar som redan var märkta på en skala 0–3 av en Bosch anställd, där värdet 0 innebar inte relevant och värde 3 innebar högst relevant. För att testa hur denna lösning kunde integreras till en webbaserad applikation gjordes en prototyp som byggde på några krav från en Bosch anställd. Därefter utförde en Bosch anställd några specifikt utformade användar-tester för att se om prototypen uppfyllde kraven.

Resultatet av de olika LLM-teknikerna visade att multi-stage pipeline presterade best i F1-score, Cohens kappa och Quadrant kappa för att bedöma artiklarna i skalan 0–3. Men i ett annat test där värde 2 och 3 sågs som positiv relevans och värde 0 och 1 sågs som negativ relevans, presterade few-shot bättre i recall av positiva artiklar. Detta innebär att few-shot hittade fler artiklar som hade positiv relevans än multi-stage. För att minimera risken att missa relevanta artiklar i prototypen hade few-shot prompt varit en bättre LLM-teknik än multi-stage pipeline.

På grund av den begränsade mängden träningsdatan i RAG 1-tekniken kan ingen trovärdig slutsats dras från resultatet för RAG 1, och eftersom RAG 1-tekniken är lik few-shot tekniken användes RAG 1 i prototypen, trots att few-shot presterade bättre i testerna. För att kunna ta ett bättre beslut om vilken LLM-teknik som ska användas i prototypen skulle mer testdata kunnat användas för att jämföra de olika LLM-teknikerna. Den slutliga prototypen testades av en Bosch anställd enligt användartesterna och resultatet visade att prototypen var lätt att använd och hade ett högt antal relevanta artiklar.

Nyckelord
Artificiell Intelligens, Stora språkmodeller (LLM), GPT-4o mini, OpenAI, Prompting, Retrieval Augmented Generering, Informationsinhämtning, Webbutveckling.

Acknowledgements

This bachelor's thesis would not exist without the support and guidance of our supervisor at Bosch, Thomas Olsson, who gave us the opportunity to write this thesis at Bosch, as well as our supervisor as Lund University, Christin Swahn, whose expertise and help was instrumental throughout this entire thesis. We are deeply grateful to both of them for their support.

Alexander Ek & Lucas Månsson

Table of contents

1. Introduction

Bosch, a global leader in technology and automotive parts, needs efficient ways to track emerging trends and relevant news. This thesis investigates using a Large Language Model to improve Bosch's ability to find and assess technical information online. The goal is to create a web-based prototype that helps filter and summarize news, supporting Bosch's innovation and decision-making.

1.1 Background

Bosch is a large supplier of various technical solutions and services. Bosch was founded in 1886 and today employs over 429 000 workers. They are the world's leading supplier of automotive parts. They also create power tools, home appliances, security systems, as well as working with electronics, cloud computing and IoT (Internet of things). Bosch applies its expertise in sensor technology, connectivity, and artificial intelligence to develop innovative and sustainable solutions.

It is important for Bosch to be at the forefront of technological development, and a central part of this entails scouting for trends in the market, such as new releases of AI models, IoT solutions, or other technology. This means that they want to stay up to date on where technological developments are heading, as well as find interesting companies to collaborate with or possibly acquire, both around the world and in local regions.

To achieve this, they search the Internet for news articles that could be of interest, providing them with vast amounts of information. One problem, however, is that this provides them with a large amount of information, which makes it time-consuming to sort out what may be interesting and relevant. Bosch currently has access to tools to facilitate the process, but these tools are expensive and provide a lot of articles that are not of interest.

Bosch wants to, with this thesis, explore the possibility of using an LLM (Large Language Model) to improve their scouting efficiency. An LLM is software that is designed to read, understand and generate text that resembles the way humans write. It is trained on vast amounts of data, where it learns to recognize patterns in text and uses this to predict what text to generate. An LLM should be used to read through sources of information and pick out what is most relevant to Bosch. It should also be used to assist in other parts of the scouting process, such as summarizing articles. These functionalities should then be implemented in a web-based prototype. Bosch will provide access to the GPT-4o mini [1] model created by OpenAI and will be the LLM the thesis will focus on.

1.2 Purpose

The purpose of the thesis is to investigate how an LLM can support Bosch's work with the search for technical news articles. The expected result should be a web-based prototype, with an easy-to-use interface, which can both identify articles and new sources of information and determine the relevance of the information to Bosch's work.

1.3 Goals

The goals of this thesis are:

- Explore how a pre-trained LLM, specifically GPT-4o mini from OpenAI, can be used to find relevant information on the Internet and assess their relevance to Bosch.
- Design and develop a web-based prototype that uses a pre-trained LLM to find information on the Internet and automatically evaluate its relevance based on Bosch's defined criteria.
- Test and evaluate the prototype to ensure its functionality and effectiveness in practical scenarios.

1.4 Problem Description

This thesis aims to answer the following questions:

1. How can an LLM be used to find relevant news articles for Bosch and be improved in its accuracy when assessing whether an article is relevant?
2. What requirements does a web-based prototype for information gathering need to fulfill to improve information scouting at Bosch?
3. How can information scouting using an LLM be integrated into a web-based prototype?
4. How well does the prototype meet Bosch’s needs?

1.5 Motivation of thesis

This thesis was chosen to do because AI is currently a trendy subject in the tech industry and has the potential to be applied in many different areas. It sounded interesting and rewarding to learn how to use such modern technology, and it can be useful to have experience with it in a future career.

Bosch has an interest in being at the forefront of technological development in order not to be outrivaled by their competition. Being able to gather relevant information to identify technological trends helps them achieve this and ensures that their business is future proofed. Developing such a tool could also replace the expensive and defective tools that Bosch uses today, which also directly benefits them financially in the short term.

Because Bosch is already at the forefront of technological development in many areas and has many employees worldwide, the company has a large group of competent personnel. This tool helps Bosch ensure that staff can apply their skills to the most relevant technology, which benefits society as it contributes to technological development at large.

1.6 Limitations

The thesis will not include a detailed analysis of the underlying theory and technology behind LLM’s. The thesis focused on optimizing the OpenAI GPT-4o mini model as this is the model that Bosch provides and will not compare results between different models. Additionally, this thesis will not consider testing how fine-tuning could optimize the LLM.

The prototype will only be developed to find information relevant to Bosch's purposes and will not consider the usage for other companies or organizations. The prototype will be designed to

be used for finding and evaluating news articles and potentially other websites, but not scientific papers as Bosch has other tools for that purpose.

1.7 Division of Labour

Table 1.1 shows a percentage approximation of how much work was done by each person on each task.

Table 1.1 Estimated division of labour in percentage.

	Alexander Sae-Cho Ek	Lucas Månsson
Literature Study	50%	50%
Requirement Specification	30%	70%
Implementing techniques	60%	40%
Testing the techniques	50%	50%
Web-app implementation	40%	60%
User Tests	70%	30%
Report	50%	50%

2. Technical background

This chapter describes the necessary technology, techniques, and theories to understand the thesis. It also briefly describes certain tools used in the thesis.

2.1 Artificial intelligence

Artificial Intelligence, often abbreviated as AI, is when a computer or machine imitates human intelligence. It has traits that allow it to learn, reason, comprehend and create text, images, and sounds, solve problems, make decisions, and exhibit creativity.

AI refers to any program that can imitate human intelligence, but recent breakthroughs in generative AI specifically have taken the world by storm and is often what people think about when they hear AI. Generative AI is a technology that can create completely original content such as text, images, and videos. They are built using machine learning, which is when an algorithm is trained to make predictions and decisions based on existing data. [2] An example of generative AI is the Large Language Model.

2.2 Large Language Models

Large Language Models, also known as LLMs, are a category of AI models that are trained on a large amount of textual data and to predict the next word in a sentence based on the context of previous words. They are capable of understanding and generating human-like language. [3]

They are built using deep learning [4], which is a type of machine learning technique that uses an artificial neural network. A neural network consists of many layers of nodes, also called neurons, that have an associated input, weight, threshold, and output value. [5] These values are called “parameters” and can be adjusted while training the model to provide more accurate outputs. Models can contain billions of parameters that can all be adjusted during training to allow the model to capture intricacies in language. [2]

An LLM processes text by first breaking it down into smaller units called tokens, which can be words, parts of words, or individual characters. When producing an output, an LLM continuously predicts the next token in a sequence using a given token's probability based on its training, the input, and context from the preceding words. [4]

By training on billions of pages of text, LLMs learn grammar, meaning, and relationships between words. This makes an LLM capable of summarizing text content, responding to a text input, and generating new text by continuously predicting the next token in a sequence. The model can be fine-tuned to improve accuracy and reduce errors or offensive content. [2]

2.2.1 GPT-4o mini

The Large Language Model used in this thesis was, as previously mentioned, the “GPT-4o mini” created by OpenAI, as this is the model initially provided by Bosch.

GPT stands for Generative Pre-trained Transformer, which refers to the models deep learning architecture, that is the “transformer” architecture. This is a common form of deep learning architecture for LLMs, [6] first proposed in the paper “Attention is as you need” by Vaswani

et al. [7]. The 4 is the generation of GPT models from OpenAI, and “o” stands for omni, which means it can accept any combination of text, audio, image, and video as input, and produce any combination of these as output. [8]

The GPT-4o mini is a smaller version of OpenAI's flagship GPT-4o model, smaller meaning that it likely contains a smaller number of parameters. While there are no official numbers from OpenAI confirming this, a Microsoft paper estimates the parameter count to 200 billion for GPT-4o and 8 billion for GPT-4o mini. [9] Still, they allow for the same number of input and output tokens for a single request, 128 000 and 16 400 respectively. [10]

Although it is smaller, it comes at a low price, at a cost of $0.15 per 1M (million) input tokens and $0.60 per 1M output tokens, compared to $2.50 per 1M input and $10 per 1M output tokens for GPT-4o. Being smaller does come at cost of performance, the difference may not be as substantial as the pricing and parameter count differences suggest [10], and it can outperform other models [11]. The performance differences can also be mitigated through different techniques described below, while keeping prices low.

2.2.2 Temperature and nucleus sampling

An LLM has several parameters that can be changed which determine the behavior of the model. Two relevant parameters to this thesis are the “temperature” and the nucleus sampling (top_p) parameters, which determine the randomness of the models output in different ways.

When an LLM is generating and selecting the next token in the output, the LLM will generate many tokens with different probability scores. These scores are determined according to how often a specific token should be chosen based on its training.

The temperature parameter is a number from 0.0 - 2.0 that determines how often tokens with a lower probability are chosen. A temperature of 0 means that the most probable token is always chosen, which means the output will always be the same for the same input. The top_p parameter also affects randomness in a different way. It is a number 0.0 - 1.0 that decides how large a percentage of the tokens are considered at all. For example, a top_p of 0.1 means that only the top 10% most likely tokens are considered. Setting these parameters to a low value will make the LLM generate more consistent and deterministic answers. [12]

2.3 LLMs for relevance judgement in information retrieval

Information retrieval in computer- and information science is the process of finding and getting the correct information from a large data set according to the needs of a user, which is most often presented though a search query, and how this can be optimized to find the most relevant and important information to the user. The effectiveness of information retrieval systems is often evaluated using relevance labels, which is often a scale that indicates the usefulness of a document for a given query and user. [14]

Previous research has explored the usage of LLMs in information retrieval, with the goal of assessing the relevance of found documents to a related search query by producing a relevance label. Faggioli, G. et. al. found that LLMs can be useful for helping humans judge a documents relevance, providing a spectrum of LLM involvement, but concluding that they might not be trustworthy as a sole evaluator since relevance is subjective and can even change over time for the same person. They concluded that the best course was to use humans as quality-control for the LLM. [17]

On the other hand, Joos, L. et. al found LLMs very efficient at filtering out irrelevant articles during a systematic literature review of scientific articles, also providing reliable evaluations, i.e. evaluations that humans agree with. [16] Thomas, P. et al. tested LLMs relevance judgements by allowing them to generate relevance labels on documents related to a search query, where a human had previously given a label that was deemed “correct”. They found that LLMs performed as good, or better, than any group of human assessors. They also provided key insights into how the LLM was prompted, and how different aspects of the prompt could influence the results. [13]

2.3.1 The multi-stage pipeline

Schnabel et. al. found that using a pipeline of two models working together could be used to further increase the accuracy of the relevance labels produced by LLMs. In these experiments, one model was first given a document and prompted to evaluate if it is relevant or not, providing a binary “yes or no” answer. The documents labeled “yes” would then be passed to another model (or other instance of the same model), which then would give it a more specific label. Using this technique, GPT-4o mini produced better results than using a single GPT-4o model, at a much lower cost. [14]

2.4 Prompting

Prompting, or prompt engineering, refers to the design of inputs to Large Language Models, and how these inputs can be optimized to guide their responses. It involves instructing an LLM to customize its behavior and enhance its capabilities. A prompt serves to control the LLM’s output and the interactions by defining specific rules and guidelines. By creating a clear prompt, the context for the conversation is set, and the LLM is instructed on what is important, as well as the desired format and content. The quality of the prompt is directly tied to the quality of the LLM's output. However, it is important to be aware that LLMs can sometimes generate incorrect answers or "hallucinate" information. The patterns may need to be adjusted as the model evolves and improves [18].

2.4.1 Zero-shot prompting

The simplest and most natural way of prompting is called zero-shot prompting (also called zero-shot learning). This is when no concrete examples are provided for the LLM to follow and relies entirely on the model’s previous training and does not add to the model’s knowledge or context [19]. Still, zero-shot prompts can be powerful when they are engineered well. For example, using LLMs with zero-shot prompting has previously been shown to produce results as reliable as humans when evaluating relevance for search-engine results from Bing. [13]

2.4.2 Few-shot prompting

Few-shot prompting is when a prompt is input to the LLM along with several examples of how the model should respond (only one example is called one-shot prompting), providing guidance which strives to improve the quality and accuracy of the output. [20]. While it is a useful prompt engineering tool that can significantly improve performance [21], the success and usefulness of few-shot prompting can vary depending on the task [22].

2.4.3 Chain-of Thought (CoT) prompting

Chain-of-thought (CoT) prompting is when the LLM is prompted to think through and break down a problem into steps, solving each step before providing the definitive answer, which can improve the performance of LLMs in logical reasoning.[20]

CoT can be implemented like a few-shot prompt, and the examples provided show the LLM how a problem should be broken down into steps, which leads to an increase in performance. [23]. A zero-shot CoT can also be used to elicit reasoning and improve the output from the model by simply adding “Let's think step by step” to the prompt, and displaying the steps, without any direct example [20]. A setback with this is that the user needs to know what reasoning steps should be taken for a specific problem.

2.4.4 Prompt patterns

A prompt pattern is a structured pattern that can be modified and adapted to fit into a specific context of a task when prompting an LLM. By adjusting certain variables within the prompt pattern, it is possible to reuse the pattern where the task is the same, but the topic or context changes. [18]

To create an effective prompt pattern for an LLM that should classify documents, there are two pattern categories that it is important to have in mind which are context control and output customization. Context control is about ensuring that the LLM receives the proper context and relevant information to generate accurate and useful outputs. Context control includes a description (explaining the intention behind the user's query), narrative (add more detail about what is relevant and irrelevant), and aspects (practical criteria that the LLM will follow). Those pattern features have the best results in document classification considering the κ and MAE scores [13]. The output customization category focuses on constraining the structure or form of the output that the LLM generates. It could be instructions that the output should be a summarization, a JSON structure or whatever the user expects the output to be.

2.5 Retrieval Augmented Generation (RAG)

Since the knowledge that an LLM has comes from data it has previously been trained on, it cannot be easily updated or expanded which can lead to it becoming outdated. This can also lead to issues with hallucinations, which is when an LLM presents incorrect information as if it were factual.

Retrieval-Augmented Generation (RAG) is a technique that improves LLMs by retrieving information from an external database. Documents from the database related to the prompt given by the user are then added as context to the LLM which, along with the user's prompt, uses them to generate a response. This helps LLMs produce more accurate answers that align more with the context and has become important for LLMs to handle more complex and knowledge-specific tasks. [24]

Documents are stored in a database by first being segmented into chunks, which are later encoded into vector representations and stored in a vector database. They are stored in a vector database because LLMs represent data (like text) in numerical form, which is called an embedding. These embeddings can then be mapped to vector representations, which include semantic features of the data. Semantic differences are then represented by the mathematical distance between two vectors, which is what allows the most relevant documents for the

prompt to be fetched. Finally, these documents are sent to the LLM to act as context for the prompt to generate an answer. [25]

2.5.1 Embedding

Embedding is a method to convert a set of text to a numerical vector (a list of numbers) which the computer can understand and process. These vectors are designed so that texts with similar meanings, like “dog” and “puppy”, are placed close to each other in a high-dimensional space. This makes it possible for a computer to compare texts based on meaning, even if the exact words are different. [26]

To find similar pieces of text, the computer compares their vectors using similarity measures. One common method is cosine similarity, which measures the angle between two vectors in a high-dimensional space. The idea is that if two texts have similar meanings, their vectors will point in roughly the same direction, resulting in a cosine value close to 1. If they are very different, the cosine value will be closer to 0. Mathematically, cosine similarity between two non-zero vectors P and Q is calculated as shown in (1):

$$Similarity = \cos(\theta) = \frac{\sum_{i=1}^{n} P_i Q_i}{\sqrt{\sum_{i=1}^{n} P_i^2} \cdot \sqrt{\sum_{i=1}^{n} Q_i^2}} \quad (1)$$

Where P and Q are the embedding vectors.

A commonly used embedding model is OpenAI's text-embedding-ada-002. This model is trained on a large amount of text data and converts a set of text to a vector with 1536 dimensions. The vector that text-embedding-ada-002 produces shows good results in use of semantic search and text classification. [27]

2.5.2 Case-Based Retrieval Augmented Generation

There are different types of RAG methods, each suited to different tasks. For the use of classifying articles, Case-based RAG is commonly used because it retrieves previous examples of how texts have been classified. When evaluating a new article, the system looks for similar articles in a database, retrieves their classifications, and uses them as a reference. This method helps the LLM to make more qualified decisions and ensures that the LLM is consistent and aligns with past decisions. Case-based RAG has been successfully applied in legal text processing, where retrieving relevant past cases improved text classification accuracy. [28]

2.6 Technology stack

This section briefly describes the primary technologies, languages, tools, frameworks, and libraries used when developing the prototype.

2.6.1 Azure OpenAI Service

To get access to the OpenAI LLM, an API (Application Programming Interface) was used. An API is, briefly, a way to integrate functionality, data, and services from another piece of software without having to implement the software from scratch [29]. In this thesis specifically,

the API from Azure OpenAI Service [30] was used to get access to the GPT-4o mini models' functionality.

Azure OpenAI Service is a part of Microsoft Azure [31], which is a cloud computing [33] platform that gives businesses access to different data services through a global network of datacenters provided by Microsoft, without having to maintain their own on-premises infrastructure, and is heavily used at Bosch R&D Center Lund.

OpenAI themselves also provide access to their models through their own API, but this was not used directly in this thesis. [32]

2.6.2 Back-end

To develop the back-end (server-side) of the prototype, Python was the language used. Python is a high-level, multi-paradigm, general-purpose interpreted programming language, and was chosen for its simplicity and ease of use, which makes it useful for developing a wide range of applications. [34] Specifically, the Anaconda Python distribution was used, which is primarily made for data science and machine learning. [35] The main features from Anaconda utilized were the Conda package manager, its virtual environment management system, and the NumPy package [38]. It was primarily chosen because it was the distribution provided by Bosch.

For building the API, a framework called FastAPI [36] was used due to its development speed and simplicity. Additionally, for dependency management, Poetry [37] was selected. Both these tools supported rapid development while maintaining a clean and efficient workflow.

2.6.3 Front-end

Since the prototype was web-based, HTML, CSS, and TypeScript were the languages used to develop the front-end. TypeScript is a statically typed version of JavaScript, and was chosen for its strong type-checking, better code maintainability, and improved developer experience in larger projects [39]. More specifically, the front-end library React [40] was used.

For styling and layout, Bootstrap CSS was used. Bootstrap is a front-end framework that includes pre-designed components and responsive grid systems, enabling simple and fast UI development and consistent design. [41]

2.6.4 Database technologies

The prototype used two different databases with different purposes. For storing simple, structured, information about users, projects, and articles, SQL Server was used [42], which is a relational database management system created by Microsoft.

To store information to be used in the RAG-setup, Qdrant [43], an open-source vector database optimized for similarity search was used to store vector embeddings. Qdrant performs well in storing and querying large vectors, which is used for semantic search.

3. Method

This chapter describes the work process of the thesis, and which methods were used in different parts of the thesis, also discussing why certain decisions were made. The work can be broadly divided into the following areas:

1. **Information gathering** – Finding information about LLMs and different techniques and how they could be used in the thesis.
2. **Requirement specification** – Interview with Bosch employee and formulation of a requirement specification, as well as finding criteria for relevant information. The specification was initially formed in a very early stage of the thesis, and later another meeting was held to revise the specification slightly after implementation had started.
3. **Implementing the LLM techniques** – From the gathered information about LLMs and how they could be optimized, how different optimization techniques were created.
4. **Testing LLM-generated relevance evaluation** – Testing how well the LLM could evaluate relevance of articles, based on the techniques described in 3.3.
5. **Implementation of the web-based prototype** – The work process of implementing the web-based prototype.
6. **Evaluating the prototype** – Evaluating the prototype by creating user tests.

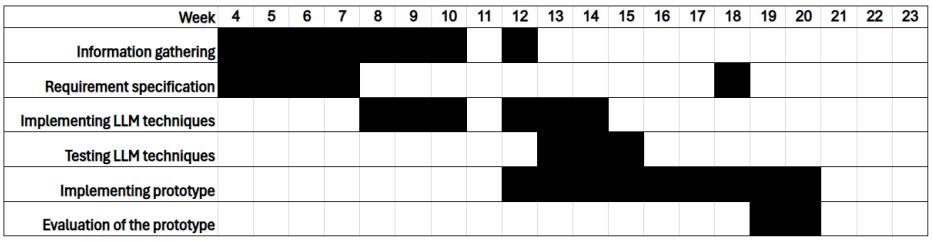

Fig. 3.1. Diagram of the work process.

The work wasn't conducted sequentially in the listed order. Fig. 3.1 describes what was worked on and when during the duration of the thesis.

The report was worked on throughout the entire duration of the thesis, which is why it was omitted from the diagram. Week 11 was exam week, which is why no work was done on the thesis there. The finals weeks consisted of finalizing the final parts of the report, creating the poster, and preparing the presentation.

3.1 Information gathering

This work involved understanding artificial intelligence and specifically LLMs, and how they could be used in this thesis. This was achieved by studying existing research and documentation on LLMs to identify techniques that could be useful for the thesis. Sources were mainly found by searching for related keywords on Google Scholar [44] and LUBsearch [45].

Initially, the search queries were broad and general, but as greater knowledge of the domain was gained, certain areas of focus were found which allowed for more specific searches and deeper knowledge. Through discussions with the supervisor at Bosch, a brief understanding of the thesis was clarified, and it was decided to focus on how an LLM can be customized to gather and assess the relevance of information for Bosch. As it was explored how such AI

could be built, certain concepts were mentioned which led to further research on how to develop the AI model.

Notes for articles and links were initially copy-pasted into a Word document on OneDrive. After a large number of sources were found, with different topics, this became unsustainable and cluttered. To organize the sources and information, an excel spreadsheet was created. Here each source was categorized according to subject and prioritized according to how relevant it was likely to be for us. Then, a Word document for each subject was created, with summaries for each relevant source. The most important takeaways for each subject were summarized in these documents, which laid the groundwork for the technical background in this report and methods used in the thesis.

3.2 Requirements & criteria

To answer the first question in the problem description, "What requirements does a web-based prototype for information gathering need to fulfill to improve information scouting at Bosch?", a meeting with an employee at Bosch was set up.

This employee has many years of experience in the industry and has worked at Bosch for several years. They also worked with information scouting in their role, which means they had interest in the prototype being a success and could be considered a key stakeholder. This meeting laid the groundwork for understanding and forming the requirements that the prototype would have to meet to improve the information gathering at Bosch.

As mentioned in the background, Bosch already had a (flawed) tool to scout for information on the Internet. The first part of the meeting was a semi-structured interview [46] about the current tool, which also included a live showcase. Open-ended questions were asked that allowed for freedom in depth and scope. These questions were:

- "What is the goal of your work with this tool?"
- "How do you use this tool in your work?"
- "What setbacks does this tool have?"
- "What is good about this tool?"

The goal of these questions was to gain as much information as possible about the current tool and workflow. Smaller follow-up questions were asked to gain further understanding of the tool. A semi-structured interview was chosen as there was no previous knowledge about the tool, which hindered the ability to ask direct questions like in a structured interview, and an unstructured interview could have led to missing information. [46]

The latter part of the meeting was about the new prototype to be developed and was more of an unstructured interview or a brainstorming session where many different ideas for the prototype were discussed.

Notes were taken during both phases of the interview and later reviewed, which led to a requirement specification. This specification contains both functional and non-functional requirements, ranked according to importance. The requirement specification served as the basis for the prototype and decisions taken during its development.

3.2.1 Finding criteria for relevant information

Defining criteria for what makes a piece of information relevant to Bosch was important for creating a prompt that would increase the likelihood that found articles would be evaluated correctly by the LLM. Finding well-defined criteria would also provide something to compare and verify the LLMs output.

Several issues arose when trying to define these criteria – relevance can be very subjective and can even change for the same person on different days. It is also hard to define relevance in an objective manner. The criteria couldn't be too strict, as that would potentially lead to the LLM filtering out interesting articles, but having too loose criteria would lead to a lot of irrelevant information being presented to the user.

A database containing articles that had previously been marked as interesting and saved by Bosch employees existed and contained 1283 articles, from the 25th of May 2022 to the 6th of January 2025. Each entry in the database contained the article title and a short summary.

This database was used to find criteria by skimming through both the title and description, looking for patterns that could give insight into what made these articles interesting. In total, around 100 articles were examined in this way. This was a time-consuming process, which is why each article could not be read in detail and only a limited number of articles could be looked at. While spending more time doing this may have given slightly better insights, even after a short time, clear patterns started to emerge.

The criteria were somewhat broad, but they still provide insight into what could be considered interesting and relevant to Bosch in the context of this thesis and were used when prompting the LLM to provide context.

3.3 Testing LLM-generated relevance evaluation

To test how well the LLM performed in evaluating articles, a test was conducted where an LLM was prompted to evaluate the relevance of found articles and give them a "relevance score". In this section, the process for how the tests were done is described.

3.3.1 Measuring model

The 0-3 scale

To compare the different LLM techniques and decide which one performs the best, a rating scale would have to be used. The rating scale is based on ordinal data which includes a score that follows a four-point scale. This rating scale could give a more detailed analyzation on the relevance assessments, while the criteria for each score could still be relatively well-defined:

- 3 – Perfectly relevant: The article is highly relevant and interesting to the employee, to such an extent that they would like to share it with other colleagues at Bosch.

- 2 – Highly relevant: The article is still relevant and interesting to the employee, but not so much that they would like to share it with other colleagues at Bosch.

- 1 – Somewhat relevant (/Related): The article may be just somewhat relevant, interesting to a minimal extent, or just related to Bosch areas of operation. The employees would rather not waste their time reading this article.

- 0 – Irrelevant: The article is not at all related to Bosch's areas of operation or doesn't contain anything that would be of interest.

This 4-point scale is inspired by relevance assessment methods used in the TREC 2019 Deep Learning Track and other research studies on document evaluation [47][14]. It was also discussed and agreed upon with the supervisor at Bosch. Additionally, it provided a clear criteria for the test subjects.

The binary scale
The scale above presents a clear divide between 0-1 and 2-3. 0 and 1 are scores given to articles that an employee would rather not waste their time reading. 2-3 represent articles that an employee wants to read. This means that, to meet the goal of "designing a web-based prototype using an LLM to gather relevant articles", only articles that got the score 2 and 3 would be displayed to the user in the prototype, and not the articles that got the score 0 and 1.

Therefore, the articles could be divided into binary scale, with a positive (score 2 and 3) and negative (score 0 and 1) category regarding the articles that will be displayed in the prototype. Table 3.1 shows how the different scores were categorized.

Table 3.1. How the articles were categorized into positive and negative.

Relevance Score	0	1	2	3
Binary Score	Negative		Positive	

3.3.2 Test process

During the information gathering phase, the techniques described in the technical background were found that could be used to evaluate articles and their relevance using an LLM. Several also seemed to show potential to optimize the LLM for this specific purpose by providing examples. To test how well the LLM could evaluate relevance, a test was conducted. Five different techniques were tested and compared. These were zero-shot prompting, few-shot prompting, multi-stage pipeline and two variations of RAG (specifics of how they were implemented are described in 3.5). The test served several purposes:

- Explore if accurate relevance scores could be produced using zero-shot prompting.
- Test and change the zero-shot prompt iteratively to see how it could be improved to get better results.
- Explore if providing examples to the LLM though a few-shot, multi-stage pipeline, or a RAG setup could produce better relevance scores compared to zero-shot.
- Give an indication of which LLM technique would perform the best when used in the prototype.

The tests were run with a temperature setting of 0. This was to ensure that any randomness caused by a higher temperature would not interfere with the test results. Since the temperature setting is 0, the setting of the top_p parameter will not have an impact on the model. The final goal of these tests was to find the most effective implementation to later be used in the prototype.

3.3.3 Test-set and training-set data

In total, four employees evaluated and scored a number of articles to be used in the tests. One of them evaluated 154 articles, one evaluated 36, and one evaluated 24, which led to a total of 214 articles being scored. All three of these employees have years of experience and domain knowledge and have information scouting as part of their job.

These labels were deemed to be “correct” and provided a test set to compare them to the LLM-generated labels. Since the few-shot and RAG techniques required some predetermined example articles with relevance labels, 40 of the 214 articles were picked out to be used in what was called a “training-set”. The 40 were chosen by picking 10 random articles of each score, using a random number generator. This was to mitigate any potential bias introduced by manually selecting articles.

This led to a set of 174 articles that the different techniques would be tested against, called the “test set”, and a set of 40 articles that would potentially be used as examples in the RAG and few-shot - the “training-set”. The breakdown of how many articles of each score were in the test-set and training-set is described in table 3.2.

Table 3.2. How articles were divided into the training– and test-sets.

Relevance score	0	1	2	3	Total
Positive/Negative	Negative		Positive		
Test set	54	71	20	29	174
Training set	10	10	10	10	40
Total	64	81	30	39	214

A larger test set would provide more data to draw a conclusion from, but since the evaluation of the articles by the employee would take a lot of their time, a trade-off considering the number of articles had to be made, which is why 214 articles were tested. There is also a large mismatch between the number of articles for each score. It was considered to cut down the number of articles for the other scores to have the same number for each, but in the end, it was decided that a greater number of articles would provide better insights, and that the results would have to be measured using statistics that take this into account.

3.3.4 Measuring the results

The relevance score that the LLM produced for each technique on each article was compared to the label that the article had in the test set. To assess which technique provided the most accurate relevance labels, the agreement between the LLM and the test set was measured using three statistics. This was calculated both for the 0-3 scale scores, and the binary scores. The metrics were calculated using the scikit-learn Python library. [48]

Cohen's Kappa

The first statistic used was Cohen's Kappa, κ, which is a robust statistic useful for interrater reliability testing since it accounts for the possibility of chance agreement. The equation Cohens Kappa is shown in (1):

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \tag{2}$$

P_0 is the actual observed agreement and P_e is the probability of chance agreement. [49]

Quadratic Weighted Kappa, QWK

While Cohen's Kappa provides insight into how often the LLM and Bosch employee agree in terms of exact matches, it does not consider how far off the LLM-generated scores are from the true values. Since, for example, the LLM rating a true 3 as a 1 would be worse than rating it a 2, larger errors should be punished more harshly. To account for this, a weighted Kappa was used, more specifically the Quadratic Weighted Kappa. [50][51]

The kappa statistics both provide a result between [-1, 1], where the number indicates the strength of the agreement according to table 3.3. [52]

Table 3.3 Approximate interpretation of kappa statistic

Kappa statistic	Strength of agreement
< 0.00	Poor
0.00 - 0.20	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.00	Almost perfect

F1-score

The second statistic used was the micro F1-score, which quantifies the model's overall performance when classifying articles, considering true positives, true negatives, false positives, and false negatives. The equation for micro F1-score is shown in (2):

$$Micro\ F1 = \frac{2 \cdot precision_{micro} \cdot recall_{micro}}{precision_{micro} + reacll_{micro}} \quad (3)$$

$$Recall_{micro} = \frac{\sum TP}{\sum TP + \sum FN} \quad (4)$$

Where $\sum TP$ is the total number of true positives over all classes, $\sum FP$ is the total number of false positives over all classes, $\sum FN$ is the total number of false negatives over all classes.

Precision and recall on each score

To identify in more detail which score the techniques had the most problem in classifying articles, the agreement between the LLM and the employees was measured in each category using two statistics.

The first statistic used recall to see how well the different techniques avoided false negatives for each category when identifying relevant articles. The equation for accuracy is shown in (4):

$$Recall\ = \frac{TP}{TP+FN} \quad (5)$$

Where TP is the number of true positive, and FN is the number of false negatives

The second statistic used precision to see how well the different techniques avoided false positives for each category when identifying relevant articles. The equation for accuracy is shown in (5):

$$\Pr e\ cision\ = \frac{TP}{TP+FP} \quad (6)$$

Where TP is the number of true positive, and FP is the number of false positive

Token count
Additionally, the amount of input and output tokens used while running each test was logged. When running the RAG tests, embedding the article text into vectors also cost embedding tokens, which were also logged.

3.4 Implementing the LLM techniques

In this section, the different LLM techniques to be tested and their respective prompts for optimizing and improving the LLM are described, and how they were structured and implemented. The full, uninterrupted prompt for each case is presented in Appendix B.

3.4.1 Creating the zero-shot/base prompt

Since the main way of interacting with any LLM is through prompts, and the quality of the prompt is directly tied to the quality of the output from an LLM, a prompt had to be engineered before any testing could begin. Principles described in the technical background were used to experiment with optimizing this base prompt. On top of this, the prompt was iteratively revised after many rounds of testing, until a result was deemed satisfactory was reached and the prompt could be used as a base for all other techniques.

The prompt created is described and explained under 4.2.1 and will throughout the rest of the report be referred to as the "base prompt."

3.4.2 The zero-shot prompting technique

The zero-shot prompting technique just uses the base-prompt and adds an article that should be evaluated in the query. The zero-shot prompt test was done many times to test changes to the prompt to get the best possible results and base prompt.

3.4.3 The few-shot prompting technique

The few-shot prompting technique uses the base-prompt along with four already scored articles, one example for each score, which is the most optimal number of shots when using few-shot prompting.[54]

```
#Examples of news articles
Here is a list of a few previously evaluated articles and their given scores.
Try to understand why each article got that given score, and use it when
reasoning about future articles you evaluate.

This article was given a score of 3.
--BEGIN SCORE 3 EXAMPLE ARTICLE--
{example_3}
--END SCORE 3 EXAMPLE ARTICLE--

This article was given a score of 2.
--BEGIN SCORE 2 EXAMPLE ARTICLE--
{example_2}
--END SCORE 2 EXAMPLE ARTICLE--

This article was given a score of 1.
--BEGIN SCORE 1 EXAMPLE ARTICLE--
{example_1}
--END SCORE 1 EXAMPLE ARTICLE--

This article was given a score of 0.
--BEGIN SCORE 0 EXAMPLE ARTICLE--
{example_0}
--END SCORE 0 EXAMPLE ARTICLE--
```

The articles used as examples were randomly chosen and were the same four for all articles evaluated by the LLM in the test.

3.4.4 The multi-stage pipeline

As described in 2.3, Schnabel et. al. found that using a pipeline of two models working together could be used to further increase the accuracy of the relevance labels produced by LLMs, increasing the results of the GPT-4o mini to better than that of the GPT-4o. [14] As GPT-4o mini was the model used in this thesis, this technique was also tested.

The prompt in this case was split up into two – one to be used as input to the first instance of the model, and one to be used in the second stage. The prompts were fairly similar, but the main difference being the scoring system used.

The first stage only gave a score of 0 or 1, where 0 represented the article as being irrelevant and 1 at least somewhat relevant. Only the articles that got a 1 in the first stage were sent to the second model that evaluated from 1-3. Exactly how these prompts were formed can be seen in Appendix B.

3.4.5 RAG 1 & 2

The RAG techniques to be tested also used example articles in the prompt, but instead of randomly picking out four static articles to be used for all tests, the RAG solution dynamically fetched the articles most similar to the article being evaluated.

To achieve this, articles in the training set were stored in a database along with its embedded vector-representation. Database articles were embedded with the “text-embedding-ada-002" model [27], which has a cost of $0.0001 per 1000 tokens embedded [30]. It was chosen because it was the embedding model provided by Bosch Lunds instance of the Azure OpenAI services. To retrieve the most similar examples, the article to be evaluated was first embedded. It was then compared to the embedded version of the example articles in the database using cosine similarity, which was calculated using methods from the NumPy library. Two RAG tests were conducted, each with slight differences in which articles were fetched.

RAG 1
The 4 most similar articles according to cosine similarity, irrespective of score, were retrieved and used as context with their scores in the prompt:

```
#Examples of news articles
Here is a list of a few previously evaluated articles and their given scores.
Understand why each article got that given score and use it when reasoning
about future articles you evaluate.

--BEGIN EXAMPLE LIST--
{example_list}
--END EXAMPLE LIST--
```

RAG 2
One article of each score was retrieved according to the highest cosine similarity for that score. The prompt was structured like the few-shot prompt.

3.5 The web-based prototype

When implementing the prototype, the requirement specification described in 4.1 and Appendix A served as the basis for what functionalities were implemented. The general goals of the implementation were that a user could create different research projects, where relevant news articles and some information about them would be presented.

3.5.1 Design and architecture

To design the web-based prototype, a classic three-tier architecture is used to keep the application structure simple since the most important part should be to create the LLM evaluation functionality. [54]

Data tier
Uses Microsoft SQL server to store the data. The database includes four tables which is described in fig 3.2:

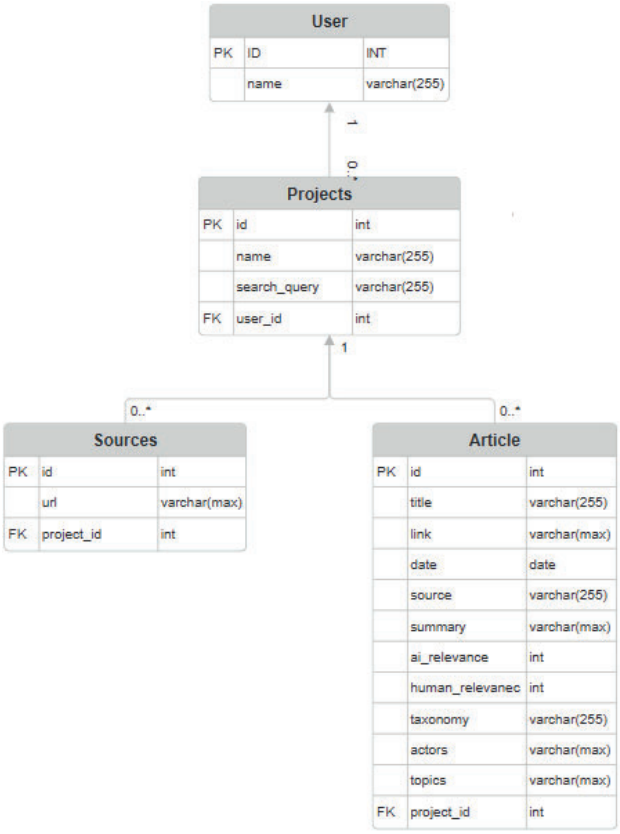

Fig. 3.2 UML Diagram of the SQL database

The “user” table was just to have basic user support, where a user can have their own projects, but largely many user-related functionalities such as authentication were ignored within the scope for this thesis. The “projects” table was to have each project, with its search query, separate from each other. The “sources” table was to store all the sources to scrape articles from for a specific project. The “articles” table was to store information and meta data about each article found and evaluated by the LLM.

The web-app used a Qdrant vector database to store embeddings of articles used in the RAG. It was chosen as it provides up to 1 GB of free cloud storage during development, and a simple and intuitive API, which made it optimal for this thesis.

Application tier
Implemented in Python with the FastAPI framework and applies the LLM technique with best result according to 4.2. The LLM was used to filter out irrelevant articles by deleting articles which got scores of 0 and 1 and displaying only articles that got 2 and 3 as scores. Also supports basic CRUD operations to the database such as creating projects, fetching news articles and saving them to the databases.

Presentation tier

Implemented using the React library and Bootstrap CSS and was designed according to the requirement specification. The design of the presentation was kept simple since the focus was placed om the functionality of the prototype and not the aesthetic design.

3.5.2 Fetching news articles

An important part of the prototype was fetching news articles to be evaluated by the LLM. This was achieved in two different ways:

The first was by retrieving news articles through a SERP (Search Engine Results Page) API. For this Serper [55] was used, which is a service that provides search results from Google Search Engine, including Google News, in a structured dictionary format, which makes it easy to work with. Here, many alternatives were looked at, but Serper was chosen as it was the cheapest option that could be used in a production environment.

Another way to fetch articles was to input the URL to a specific news site – the prototype would then find all links on the webpage and use an LLM to filter out links that were likely to lead to a news article. These articles would then be scraped.

To scrape the article content, an HTTP GET request was sent to the article's URL. The webpage was then parsed with the Python library BeautifulSoup [56] to extract article content, and other data such as date, title, and link, along with the module “re” from the standard library.

When scraping content from the internet, it is important to pay respect to the websites robots.txt file, which defines rules for bots, such as web crawlers, and the terms of use for the website. Also, full article contents were not saved long-term, only the embedded version of the article content, and metadata for the article such as the link.

3.5.3 LLM integration

The idea for the LLM integration was that, after fetching a large number of news articles from the internet, the LLM would evaluate each article according to the relevance to the user and Bosch, and then only the articles deemed relevant (positive) would be presented to the user, as was briefly discussed in 3.3.1. The technique which provided the best result from the test was the one chosen to be used in this part of the prototype.

Additionally, LLM was used to summarize the information from all the relevant articles to give the user an overview of the articles, and to recommend news sources to the user based on a search query.

3.6 Evaluation of the prototype

To evaluate the prototype, a User Acceptance Testing (UAT) was performed by a Bosch employee. UAT was used because the prototype needed to be tested by a real Bosch user to validate whether the system achieves the requirements that were previously outlined. [57]

The employee was prompted to test out the features of the app, focusing on the “Project” page where the main part of the app's functionality was located. Feedback from the UAT could be divided into two parts, where the first part concerned their general user experience. The following questions were asked:

1. How was your general experience using the app? Was something hard to understand or use?
2. Do you get enough information about the articles that are represented on the website?
3. Is the layout and general design satisfactory?
4. Would you say that the articles presented were relevant to you?
5. How many would you say were relevant?

After this, the requirement specification was reviewed together with the employee to validate which requirements were fulfilled.

3.7 Source evaluation

The sources used in this report are all from authorized providers but vary from scientific reports to companies' technical information. In this section, the sources are grouped into similar source types, and their trustworthiness is evaluated.

Company websites
Sources: [1], [2], [3], [4], [5], [6], [8], [10], [12], [19], [27], [29] [30], [31], [32], [33], [42], [54], [61],[62],[64]
These sources come from famous technological companies (OpenAI, IBM, Microsoft, DocsBot AI) which means that they themselves control the information. This increases the risk for biased content but since the companies are famous, the content would be criticized if their content is false or biased. Since they are at the forefront of technology, they have good knowledge withing technology and their information is mostly a primary source. Overall, these sources are evaluated as trustworthy sources.

Peer-reviewed articles
Sources: [11], [13], [15], [22], [23], [26], [49], [50], [51], [52], [53], [59], [65]
These sources provide scientific articles that have been peer-reviewed before they were published. This makes the sources highly trustworthy.

ArXiv
Sources: [7], [9], [14], [16], [17], [18], [20], [21], [24], [25], [28], [47], [58], [63]
Articles that are published on arXiv have not necessarily been peer-reviewed, and the articles can vary in quality. Most of the articles used in this report show indication that they have been written with academic intention. Overall, this source is moderately trustworthy.

Technical documentation and specifications
Sources: [34], [35], [36], [37], [38], [39], [40], [41], [43], [48], [55], [56]
These are the sources from the tools that were used in the thesis. They provide documentation for their own released product which makes the sources highly trustworthy for technical reference. But since they are sources for their own product, it could be biased for other use cases than technical references.

Search engines
Sources: [44], [45]
These are sources to the search tools that were used during the thesis to find relevant useful articles for the thesis. They were not used directly but rather to find other academic work. They both provide academic articles but google scholar doesn't guarantee that the articles are peer-revied while LUBsearch provides articles from the university that must have been peer-reviewed before the articles are added to their database. The article in Google scholar must

therefore be critically evaluated which makes it moderately trustworthy while LUBsearch has more reliable academic articles which makes it highly trustworthy.

Books
Sources: [46]
This is an academic book which is published by a famous Swedish publisher, "Studentlitteratur". The book was published in 2013, which makes it old and the information in the book could be outdated. But since the information is about how to use interview as a scientific method which at that time already was widely explored topic, the chance that the information is outdated is low. Therefore, the source can be seen as trustworthy.

News-sites
Sources: [60]
The Verge is a famous technology news website, and their articles are published by reporters and journalists. But since the article is not academic and could be biased, the source is moderately trustworthy.

GeeksForGeeks
Sources: [57]
GeeksForGeeks is an educational portal that provides learning content within technology and is widely used in the programming community. The content is generally objective, but some articles are created by users of the platform and not by experts, which means that the information must be used with caution. Therefor the source would be assessed as slightly trustworthy.

4. Results & analysis

This chapter presents the results of the final requirements specification, different measurements in how the LLM performed in labeling articles, and the prompt usage for every LLM technique.

4.1 Requirement specification

This section presents the requirement specification which was created from the semi-structured and unstructured interview with the Bosch employee. It contains both functional and non-functional requirements on how the prototype should be built. The requirements are also ranked on a priority scale from 1 to 3, where 1 indicated the highest priority, that is the requirements that would have to be fulfilled to consider the prototype a success. 2 meant medium priority, requirements are functionalities that could improve the prototype but are not crucial for the users or thesis. 3 was for requirements that should be implemented last if there is time. The result of the requirement specification contained 25 requirements where 20 of them were functional requirements and 5 of them were non-functional requirements. Here are the most important functional requirements and non-functional requirements for the prototype:

FR1.1 The prototype should find news articles on the internet related to a search query.

FR1.2 Found articles should be evaluated by an LLM according to their relevance.

FR1.3 The prototype should present news articles to the user in a list, with each entry containing the following information: Title, short excerpt, publication date, publisher/source, link to article.

FR1.6 The user should be able to create several search queries that are updated with new articles on a regular basis.

NFR1.1 The articles presented should be relevant to the user and related to the search query. Articles that are related to the query but not relevant to the user should be filtered out.

NFR1.2 The articles presented should be related to new technologies, technological advancements, or new tech companies. Articles regarding what Bosch's competitors are doing are also of interest. Other non-technical matters such as stock prices or organizational changes are of lower priority.

The full requirement specification can be found in Appendix A.

4.2 The zero-shot/base prompt

This section presents the final base prompt that was used in all the tests and the final prototype. It also presents the first version of the prompt, before any iterative testing, but the final base prompt was the one used in the prototype.

4.2.1 The first iteration of the prompt

This was the first iteration of the prompt, before any testing began, strictly based on and inspired by previous research.

```
Assume that an employee is looking for news articles to find technological
developments and identify technical trends for the company Bosch.

The one searching is an employee at Bosch, whose job is to scout for new
technologies, technological advancements, or trends in areas and industries that
Bosch operates in.

#Task
Given a news article, you must decide if it is relevant for the company Bosch by
providing a score on an integer scale of 0 to 3 with the following meanings:

3 = Perfectly relevant, the information is very useful to Bosch. The employee is
highly likely to find this very interesting
2 = Highly relevant, the information is probably useful to Bosch. The employee is
likely to find this interesting.
1 = Relevant, the information could be useful for Bosch. The employee might find
this interesting.
0 = not relevant. Information is not technical or would not be interesting to
Bosch. The employee is unlikely to find this interesting.

Here are some examples of article topics and content that might make an article
interesting and useful:
- Companies or organizations making technological advancements in technologies or
industries that Bosch also work or operate within
- Companies or organizations investing in technologies that Bosch also works
with.
- Companies or organizations investing in other companies - if the companies are
in the same industries as Bosch.
- Companies or organizations collaborating, if the companies are in the same
industries as Bosch.
- Organization or company roadmaps or strategy plans being revealed, in they
operate in the same industry as Bosch.
- Products/companies winning innovation prizes for a certain technology.

#Article
Consider the following news article:
--BEGIN ARTICLE CONTENT--
{article_content}
--END ARTICLE CONTENT--

#Instructions
Split this problem into steps:
1. Consider how well the content matches the likely intent of Bosch's research.
2. Consider if the article matches any of the examples previously mentioned for
what might make an article interesting.
3. Consider if the content relates to technological advancements or innovations.
4. Consider if the article contains information about Bosch's competitors or
companies that share an industry with Bosch.
5. Consider how trustworthy the source is.

Consider the aspects above and the relative importance of each, evaluate the
overall relevance and decide on a final score.

# Output format
Summarize the most important takeaways of the article. Explain your score
according to the criteria above and produce an integer 0-3. Respond in the
following format, with the following fields:
"summary": "Summary of the article, highlighting the most important take-aways",
```

```
"explanation": "Reasoning and explanation of the given score",
"score": X,
```

4.2.2 Final base prompt

The first part of the prompt describes the task for the LLM to complete. It covers the two prompt pattern features “description” and “narrative” within the prompt pattern category “context control”.

```
#Context
Assume that a user is looking for interesting news articles to find technological
developments and identify technical trends for the company Bosch.
The one searching is an employee at Bosch, whose job is to scout for new
technologies, technological advancements, or tech trends in areas and industries
that Bosch operates in.
```

In this part of the prompt, a custom query that describes the domains and areas of interest are input in place of `{topics}`. This gives the LLM context on what is relevant to the specific employee. These domains are meant to be able to be customized by a user in the prototype.

```
This employee is looking for information about {topics}.
```

The following part describes the specific task:

```
Task
Given a news article, you must decide if it is relevant for the company Bosch by
providing a score on an integer scale of 0 to 3 with the following meanings:

3 = Perfectly relevant: The article provides highly valuable information that aligns
directly with Bosch's technological advancements, strategic direction, or areas of
interest. It provides critical insights that would be of immediate interest to the
employee. Only award a score of 3 if the article presents highly actionable,
impactful, and specific insights that align directly with Bosch's current
technological focus or strategic goals.

2 = Highly relevant: The article provides significant value to Bosch but is not as
immediately actionable as a score of 3. It still addresses important technological
trends or competitive insights within Bosch’s industries. The article is still
highly useful but may not be immediately actionable. It discusses important
technological trends, major industry shifts, or competitor activities that are
relevant to Bosch but not groundbreaking.

1 = Relevant: The article has some relevance, such as mentioning technological
trends or innovations, but it’s not a direct fit with Bosch’s current strategic or
technological focus. It may be of moderate interest, but likely doesn’t directly
impact Bosch's immediate priorities. The article relates to Bosch's industries or
technologies. It might discuss trends, emerging technologies, or general industry
insights that are worth noting but provides less clear, actionable intelligence.
0 = Not relevant: The article is not as relevant. It may be non-technical, unrelated
to Bosch’s areas of operation, or not a news article at all. The content does not
provide any actionable insights that would be of interest to a Bosch employee.
Articles that are tangentially relevant or discuss trends that are only vaguely
connected to Bosch’s business should receive a score of 0. Also articles that are
unrelated to Bosch’s technological and business focus, or for content that is non-
technical, untrustworthy, or irrelevant.
```

The following part of the prompt attempts to guide and help the LLM in its evaluations by providing the criteria described in 3.3.

```
#Examples of interesting topics and content
Below are examples of article topics and content that would be relevant, interesting
and useful for the employee and Bosch:
- Companies or organizations making significant technological advancements in
technologies or industries that Bosch is involved in.
- A company or competitor releasing a new product that could compete with Bosch.
- Companies or organizations investing in technologies or industries relevant to
Bosch.
- Companies investing in or acquiring other companies that are in the same industries
as Bosch.
- Collaborations or partnerships between companies that operate in Bosch's
industries.
- Roadmaps or strategy plans from companies or organizations that are relevant to
Bosch’s industry or technological focus.
- Companies or products winning innovation awards or prizes related to Bosch’s
technological interests.
```

The following part of the prompt provided the article contents to be evaluated.

```
# Article to be evaluated
Consider the following news article:
--BEGIN ARTICLE CONTENT--
{article_content}
--END ARTICLE CONTENT--
```

The following part of the prompt described the aspects that the LLM should consider when evaluating relevance. Here, a form of chain-of-thought prompting is used, since the overall task of evaluating the relevance of the article is broken down into several steps which then are all considered when producing the final score.

```
# Instructions
When evaluating an article, consider the following steps:

1. Does the content align with Bosch's technological priorities and goals? Does it discuss technologies, industries, or competitors directly related to Bosch’s business?
2. Is the article about a significant, cutting-edge technology, trend, or innovation that could impact Bosch's operations? Only advancements that are clearly actionable or strategic for Bosch should score highly.
3. Is the article about a company or organization within the same industry or field as Bosch? Does it mention companies that directly compete with or complement Bosch’s technology?
4. Does the article provide actionable insights or valuable intelligence? Is it something that could directly impact Bosch’s decision-making process, technological strategy, or competitive standing?
5. Can the article be categorized into one of the topics previously presented?
6. Does the article come from a reputable, reliable source? Prioritize content from credible sources that are known for reporting on technological trends and industry advancements.

Consider the aspects above and the relative importance of each, evaluate the overall relevance and decide on a final score. Be pessimistic when evaluating, that is opt for a lower score if unsure.
```

The following part covers the prompt pattern category “output customization”, by telling the LLM what output it should generate. It was asked to provide a summary so that, in case strange evaluations appeared, it could be double checked that the LLM got the right content from the web page. It was also asked to provide an explanation to elicit reasoning and make use of the chain of thought prompting, since the GPT-4o mini is not a reasoning model.

```
# Output format
Summarize the most important takeaways of the article. Reason and explain your score according to the criteria above and produce an integer 0-3. Respond in the following format, with the following fields:
"explanation": "Reasoning and explanation of the given score",
"score": X
```

4.3 Relevance evaluation tests

This section will present the results of comparing the LLM-generated relevance labels to the ones given by the employees according to the metrics described in 3.3.4. Table 4.4 shows a summary of the results, which indicates the overall performance of the LLM in evaluating articles on the 0-3 scale. These were the numbers primarily looked at when evaluating which technique to use, along with the binary results described in section 4.3.4.

Table 4.4. A summary of the 0-3 scale results.

	F1-Score	Cohens Kappa	QWK	Average cost per article ($)
Zero-shot	0.425	0.179	0.611	**0.000423**
Few-shot	0.437	0.211	0.581	0.000936
Multi-stage	**0.5**	**0.312**	**0.663**	0.000528
RAG 1	0.477	0.258	0.631	0.001077
RAG 2	0.402	0.144	0.558	0.001087

4.3.1 Results for each score 0-3

The precision, recall and micro F1-score for each technique in total and for each relevance score was calculated as is shown in table 4.1.

Table 4.1. Results of all techniques in evaluating articles in terms of precision, recall and F1. score.

Technique	Score	Precision	Recall	F1-score
Zero-shot	0	0.818	0.167	0.277
	1	0.460	**0.648**	**0.538**
	2	0.122	0.300	0.173
	3	**0.928**	0.448	0.604
	Total	-	-	**0.425**
Few-shot	0	0.867	0.241	0.385
	1	0.472	0.591	0.525
	2	**0.148**	**0.400**	**0.217**
	3	0.813	0.448	0.577
	Total	-	-	**0.437**
Multi-stage pipeline	0	0.608	**0.833**	**0.703**
	1	**0.674**	0.408	0.508
	2	0.122	0.300	0.173
	3	0.875	0.241	0.377
	Total	-	-	**0.500**
RAG 1	0	**0.913**	0.389	0.546
	1	0.477	0.592	0.528
	2	0.093	0.200	0.127
	3	0.800	**0.552**	**0.653**
	Total	-	-	**0.477**
RAG 2	0	0.833	0.278	0.417
	1	0.454	0.620	0.524
	2	0.061	0.150	0.087
	3	0.800	0.276	0.410
	Total	-	-	**0.402**

Zero-shot

Zero-shot prompting achieves high precision for articles with 0 and 3 as actual score, indicating high reliability when predicting these scores. However, its low recall for class 0 (0.167) suggests that the model is highly conservative in giving an article score of 0 and fails to identify many truly scored 0 articles. The recall had the highest result (0.648) in articles with score 1, but the precision in score 1 was a bit lower (0.460) meaning that zero-shot was likely to give out score 1 to most of the articles.

For articles with 2 as true score is notably weak across all metrics, particularly with a low F1-score of 0.095, indicating a significant challenge in identifying moderately relevant content. Performance for articles with 3 as actual score is great in precision, the best of all techniques, but not quite as good in recall.

Few-shot

Few-shot prompting showed slightly more balanced results compared to zero-shot. Both precision and recall increased for score 0, resulting in a higher F1-score (0.385). It had the best

precision and recall for score 2 of all tests, resulting in the highest F1 for score 2, 0.217, although this is still not that great of a number.

The few-shot examples were chosen at random when running the tests, which could of course impact the score. It is possible that better results could be obtained if articles were chosen carefully. If implemented in the prototype, either the user would have to choose the articles to be used as examples, or they would have to be fetched dynamically, which would be like the RAG 2 solution that decreased performance.

Multi-stage pipeline
The multi-stage pipeline showed the overall best performance, getting the highest total F1-score. For score 0 it had by far the highest recall, but the lowest precision which indicates that this technique gave out score 0 more often than other techniques. Despite this, it had the highest F1-score for the score of 0 also. Its performance for the score 3 is weak, with an F1-score of 0.377, caused by low recall (0.241), which implies a failure to capture highly relevant articles, but still precision is high meaning that multi-stage is conservative in giving an article a score of 3, but often correct when doing so.

RAG 1
RAG 1 has the best performance for articles with 3 as actual score showing high results in both precision (0.800) and recall (0.552) giving the highest F1-score (0.653). For the other scores, RAG shows overall good performance when compared to the other techniques, resulting in the second highest F1-score of 0.477.

RAG 2
RAG 2 showed the worst performance overall, with the lowest total F1-score and not getting the best for any individual score. This solution is very similar to few-shot, only that here the example articles are fetched dynamically, which makes it noteworthy that the results are worse than few-shot. It is even worse than the zero-shot prompt, which is also interesting since it is the only technique that decreases performance.

Summary of precision, recall & micro F1-score for all LLM techniques
Most techniques struggle with class 2 across all metrics. This could be due to a low number of training examples of articles with 2 as actual score, or just because it is the hardest score to define. Therefore, it is hard to reach a fair conclusion in this category.

Techniques that achieve high recall for higher relevance scores (classes 2 and 3), such as RAG 1 and few-shot prompting (less conservating in giving out high scores), are more valuable in the context of relevance prediction, where failing to identify relevant content is more costly than including irrelevant material, at least to a certain extent.

The multi-stage pipeline had the overall best F1-score, suggesting that it is the most effective technique in this context. This is partly because it had an especially F1-score for score 0, while maintaining good results everywhere else. The close second in terms of F1-score, RAG 1, had a high F1-score for score 3, while also performing well in other categories. An interesting step forward could be to combine RAG 1 and the multi-stage pipeline to explore if even better results could be achieved.

4.3.2 Kappa statistics

Table 4.2 presents how many articles the LLM evaluated correctly and how many times it got the score wrong by one and by two respectively. It also presents the Cohen's –and Quadratically Weighted Kappa's for each test.

Table 4.2. Results for all techniques for Kappa statistics and number of correct scores.

	Correct	**1 off**	**>1 off**	**Cohens Kappa**	**Quadratic Weighted Kappa (QWK)**
Zero-shot	74	96	4	0.179	0.611
Few-shot	76	90	8	0.211	0.581
Multi-stage	**87**	78	9	**0.312**	**0.663**
RAG 1	83	82	9	0.258	0.630
RAG 2	70	97	7	0.144	0.558

Considering Cohens Kappa, Few-shot learning, multi-stage pipeline, and RAG 1 provided a number that indicated fair agreement between the LLM and the Bosch employee's evaluation when accounting for perfect matches, with zero-shot and RAG 2 falling just below. Cohens Kappa only accounts for perfect matches between the ratings, which means that the LLM is generally fair at perfectly evaluating the articles, with the multi-stage pipeline being the best with a score of 0.31. That means it is 116,7% better than the worst technique, RAG 2, and 74,3% better than zero-shot.

Considering the Quadratic Weighted Kappa indicates that the LLM provides substantial agreement, which means it does not make many errors where it is off by more than one. By this statistic, the multi-stage pipeline also produced the best results, outperforming zero-shot by 8,5%, few-shot by 14,1%, and RAG 1 by 5,2%. RAG 2 is the worst technique by this metric also. For QWK, few-shot is worse than zero-shot, even though it is better considering normal Cohen's Kappa, indicating that few-shot makes less mistakes, but more large ones

While the LLM in general is at best fair at providing correct scores, as seen by Cohens Kappa, it makes very few large errors as seen by the QWK. The multi-stage pipeline and RAG 1 techniques are the best in both statistics, placing 1st and 2nd in both. This indicates that providing examples does increase the performance of the LLM. However, this is not the case for RAG 2, where performance is lower than that of even the zero-shot prompt.

4.3.3 Token usage

The cost was calculated according to the token cost of GPT-4o mini ($0.15/1M prompt, $0.60/1M output) and the embedding model text-embedding-ada-002 ($0.0001/1000 embeddings). The total amount of tokens used for all articles was divided by the number of articles, which gave the average cost per article for each technique, described in table 4.3.

Table. 4.3. The average costs per article for each technique.

Technique	Average cost per article ($)
Zero-shot	**0.000423**
Few-shot	0.000936
Multi-stage	0.000528
RAG 1	0.001077
RAG 2	0.001087

Zero-shot was the cheapest. Few-shot was around 2.2 times more expensive, multi-stage around 1.25 times more expensive, and both RAG solutions about 2.55 times more expensive. It is worth noting that the multi-stage pipeline uses slightly less tokens than that of the zero-shot prompt for the same article in the first stage, since the prompt is shorter. Only articles that are evaluated to a 1 in the first stage bring the total token usage up to more than zero-shot in the second stage.

While the differences between the techniques can seem substantial, the costs can still be described objectively as low. The previous tool used at Bosch had a yearly subscription fee of $1500 per individual user, so even using the costliest technique, 1 379 944 articles could be evaluated for the same price.

4.3.4 Binary results

Table 4.5 shows the results in terms of the metrics described in 3.3.4 when dividing the scores into negative (0-1) and positive (2-3) scores.

Table 4.5. The results of the LLM techniques for the binary evaluation.

	Score	Precision	Recall	False positives	False negatives	Cohens Kappa	F1-score
Zero-shot	Negative	0.883	0.784	27	13	0.477	0.770
	Positive	0.571	0.735				
Few-shot	Negative	**0.885**	0.736	33	**12**	0.434	0.741
	Positive	0.529	**0.755**				
Multi-stage	Negative	0.880	**0.824**	**22**	14	**0.513**	**0.793**
	Positive	**0.614**	0.714				
RAG 1	Negative	0.847	0.752	31	17	0.372	0.724
	Positive	0.508	0.653				
RAG 2	Negative	0.843	0.776	28	18	0.385	0.736
	Positive	0.525	0.633				

When implementing the LLM techniques into a web application, the articles with score 2 and 3 will be shown and the articles with score 0 and 1 will not be shown, which means that the articles can be divided into binary categories which can be called a positive and a negative category. The result of this binary splitting shows that all the LLM techniques perform quite similarly where the best results are from multi-stage pipeline with 0.793 in F1-score. The multi-stage pipeline also performs best in having the smallest number of false positives, but the few-shot technique has the smallest number of false negatives.

When looking at precision and recall, the multi-stage pipeline has the best recall in negative articles, meaning it is good at finding all the negative articles. Few-shots have the best recall of positive articles meaning that few-shot don't miss positive articles. Few-shot also has the best precision in finding negative articles meaning it is best at giving out correct score to negative articles, and the multi-stage pipeline have best precision in finding positive articles meaning it is best at giving out correct score to the positive articles.

It is worth noting that with the few-shot prompt, the articles to be used as examples were chosen at random. Although the results for the technique were good, it must be considered that the example articles used may have been particularly good for the test set, and that other example articles may have given worse results.

4.4 Prototype result

This section presents the results of the prototype design and the results of the user test.

4.4.1 Prototype design

The result of the prototype was a web-application with 4 pages which is described in fig 4.1. The overall workflow of the web app is that a user creates projects for each specific research area. In each project, articles related to a search query and from given websites are found and evaluated by an LLM. Only articles that were given a score of 2 or 3 by the LLM are then displayed to the user.

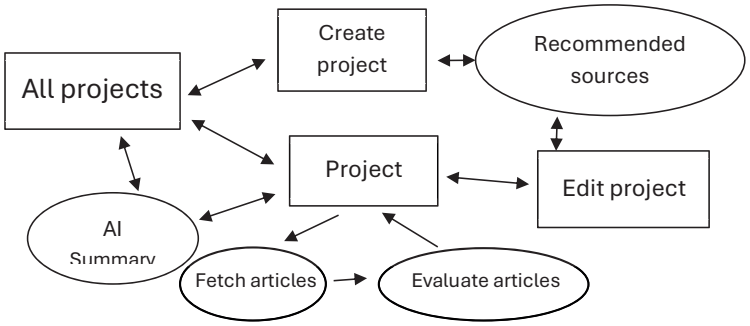

Fig. 4.1. The navigation flow diagram of the prototype where the rectangles are all the pages, and the ovals are the most important functions of the prototype.

All projects
The page called "All projects" is the homepage of the prototype where all the different research projects are shown. Here, the user can get an LLM generated summary of all articles in all projects. The GUI design of the page is shown in fig. 4.2.

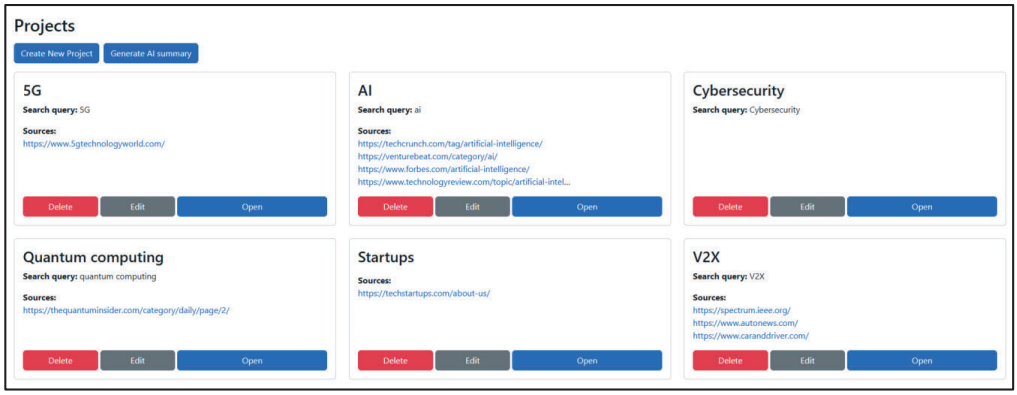

Fig. 4.2. GUI design of the "All projects" page.

A user can also get an AI generated summary for all projects. Visually, the GUI design is as shown in fig. 4.5.

Create project

From the homepage the user can create a new project where the user decides which website the prototype will get articles from, and the user can write a search query which will be used to fetch articles from unspecified websites using serper API. After creating a project, the user is sent back to the home page. The GUI design of the page is shown in fig 4.3. Here is also an option for getting recommended news sources by an LLM based on the search query. Example is shown in fig. 4.7.

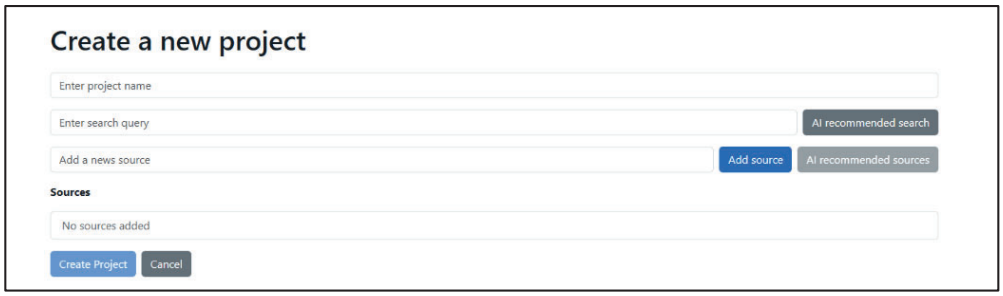

Fig. 4.3. GUI design of the "Create project" page

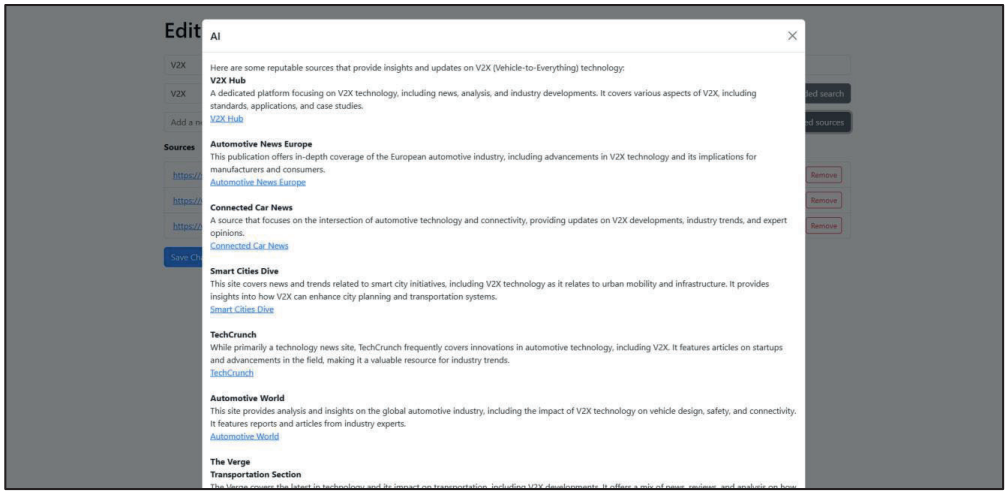

Fig 4.7. Example of the AI generated recommended sources.

Project

From the homepage, the user can also click on a project and go to the project page. In the project page the user can click on a button to fetch articles which will fetch articles from the website's sources and the search query. The fetched articles are then evaluated by the LLM technique that is implemented in the prototype and then the articles with scores of 2 or 3 are displayed on the page. When a user clicks on one of the buttons to the right, the article will be embedded and stored in the vector database along with a score, to be used in the RAG. The green button saves the article with a relevance score of 3, and the red button saves it with a score of 1. Each article shows information about the article title, link to article, date the article was published, which technological area the article belongs to according to Bosch internal taxonomy, which actors are mentioned in the article, and a short summary of the article.

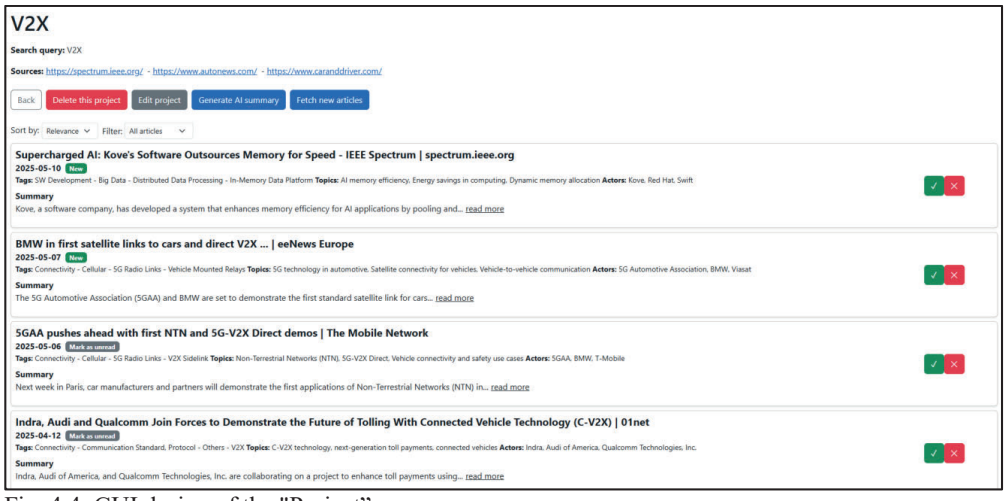

Fig. 4.4. GUI design of the "Project" page

Additionally, on this page the user can sort the articles by score and by date the articles were published. The user can also filter out old articles that the user already has read, and there is a summary function that summarizes the information of all the articles on this page. The GUI design of the page is shown in fig 4.4, and the GUI design of the summary function is shown in fig 4.5.

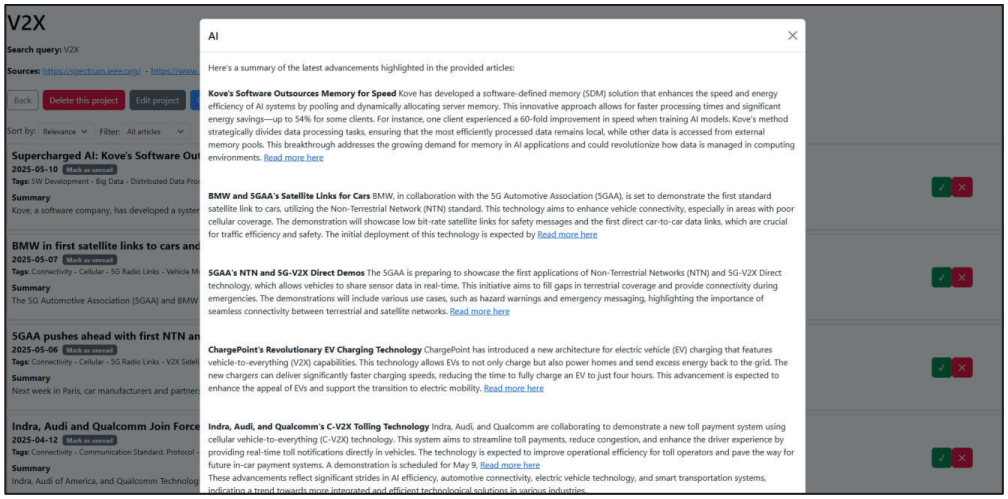

Fig. 4.5. GUI design of the summary function

Edit project
The user can also edit the project, which sends the user to the edit project page. There, the user can add more sources or change the search query of the project. The GUI design of the page is shown in fig 4.6.

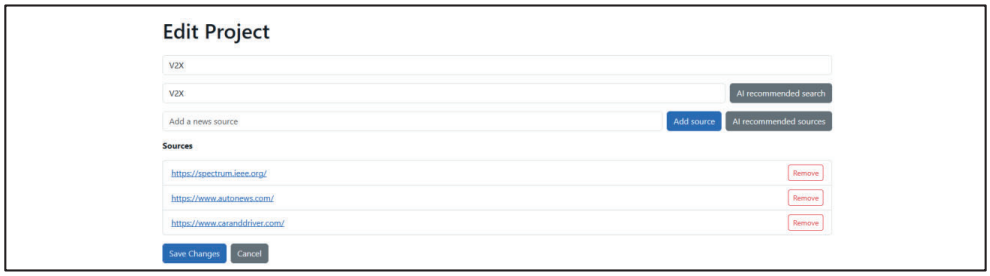

Fig 4.6. GUI design of the "Edit project" page

4.4.2 User test result

How was your general experience using the app? Was something hard to understand or use?
The prototype had clear buttons to navigate and there were no issues using the app.

Do you get enough information about the articles that are represented on the website?

The articles showed everything I needed to know to get an overview of the article. Maybe the summary text could have a bigger font-size.

Is the layout and general design satisfactory?
Make each article more compact.

Would you say that the articles presented were relevant to you?
The articles a mostly relevant, but some articles are duplicates. Also, some blog posts appear that would not be classified as news articles and therefore less interesting.

How many would you say were relevant and not relevant?
Of 30 articles I would say that 21 of them were relevant and 9 was irrelevant

Requirements
When reviewing the requirement specification, it was found that all requirements except 5 could be fulfilled. These are the requirements that were not fulfilled:

- FR2.10 The user should be able to filter out certain domains and websites.
- FR2.11 The user should be able to save articles.
- FR3.1 The prototype should, on top of news articles, also find other content on the internet.
- FR3.2 The prototype should alert the user when new articles for a certain search category appear.
- FR3.3 Occasionally, an article deemed less relevant should be presented to the user, and they should be prompted to mark it.
- NFR3.1 An LLM should take the users search query input and rework it to generate a better search query that will increase the chance to get more relevant news articles.

Two of them were of medium priority and 3 were of low priority. The reason for this was time limitations and that more emphasis was placed on completing the requirements of higher priority.

5. Discussion

This chapter discusses the result of the LLM techniques and the tests, which LLM technique is best suited for the prototype and the result of the prototype.

5.1 The LLM techniques

During the information gathering phase, many techniques to improve the LLM were found, and the 5 techniques were chosen as they were suited for the purpose of evaluating the relevance of an article. The zero-shot prompt is a technique that can be used to almost all LLM applications and is included in the other LLM techniques. Even after iterating many times, it is hard to know that the optimal zero-shot prompt is optimal, since small changes in the input can alter the output of the LLM, even just rephrasing. [13]

The few-shot prompt is suited for the thesis purpose given our results because it gives the LLM some example articles which most likely will help the LLM to understand what a relevant article is. It must be considered, though, that the few-shot examples used in the tests were chosen at random and used for all tests. It may be the case that they were particularly good examples, and that the results may have been worse otherwise. For this reason, the few-shot technique was not used in the prototype.

The multi-stage prompt could be helpful in the way that it gives the LLM a chance to analyze its own answer and probably find some possible mistakes in its answer. The RAG 1 technique is used because it will be like an improvement on few-shot prompt where the few-shots article will be about the same topic as the article that will be evaluated. The RAG 2 technique is used to see if it could improve the RAG 1 technique by having example articles within the same topic as the input article and having example articles of each score.

To improve the thesis and the survey of which LLM techniques perform the best, there are different changes that could have been made. First of all, the test- and train-set could have included more articles in general and included articles on a bigger variety of topics. If the test-set of articles had a bigger variety of topics, it would increase the chance to see if the result of the few-shot prompt will be affected by the different topics in examples articles and the input article.

If the training-set included more articles in general, the vector database for RAG 1 and 2 would have more articles to choose between, which probably would increase the chance of retrieving better articles in the prompt.

The test-set could also include the exact same number of articles for each score. In the results, the multi-stage techniques performed best overall, but when analyzing the results, the multistage was giving in general a lower score to each article. The test-set consisted of most articles with lower scores, so that could be a reason why the multi-stage was performing best overall.

The LLM techniques chosen in the thesis are the ones that were considered the most relevant to the thesis and seemed to have the most potential for the purpose of the usage of the LLM. There are many different LLM techniques and a large variety of similar LLM techniques that also could be tested, but most of them would have similar concepts to the ones used in this thesis and the results would probably be similar to the current results.

5.2 LLM Relevance labelling tests

Relevance is quite a subjective term and because of this formulating concrete criteria that the information must meet was not possible, although it was possible to find general guidelines. These guidelines were found by scanning through a database of already found relevant articles, looking for patterns. These "criteria" were later added into the input to the LLM as context.

The results in table 4.5 (binary results) show that all the LLM techniques had around 70% to 80% in F1-score, which is an improvement compared to the earlier method Bosch used to search for relevant articles. Bosch's earlier tools showed a lot of irrelevant articles and one of their employees estimated that only 5% of all the articles seemed to be relevant. Therefore, with the implementation of an LLM as a relevance evaluator used to filter articles, they will save a lot of time reading irrelevant articles.

Although using an LLM is a big improvement, the different LLM techniques don't seem to differ that much where RAG 1 had the lowest F1-score at 72.4% and multi-stage pipeline had the best F1-score at 79.3%. Although the LLM techniques have similar F1-scores, they differ a bit in precision and recall, which can be an important aspect when deciding which techniques should be implemented into the web-application. Multi-stage pipeline, which had the best F1-score, had the best recall in negatives articles, while few-shot and zero-shot have the best recall in positive articles. This means that both few-shot and zero-shot are better at finding all the positive articles while there is a bigger chance that the multi-stage pipeline misses a positive article. Missing a positive article could lead to Bosch missing an article with important information.

While the few-shot prompt showed that providing examples to the LLM could increase performance, since our few-shot tests used random articles it is hard to say that they were just not "lucky" with those specific examples. For this reason, the RAG 1 technique was chosen to be used in the prototype, since it allowed for dynamic fetching of articles and provided good results.

5.3 The prototype

The prototype resulted in an easy-to-use, full-stack, web-based application. The Bosch employees found it a useful tool for scouting news since it provided mostly articles that were considered relevant to Bosch. Unfortunately, there was only time to complete one user test for the prototype within the scope of this thesis. This is a clear area for improvement.

Another improvement of the prototype could be to evaluate the articles that weren't displayed on the prototype. To test if the prototype showed the most relevant articles, an evaluation could have been done by Bosch employees were they could have assessed how many relevant and irrelevant article there were in the articles that were showed in the prototype and how many relevant and irrelevant articles that weren't showed in the prototype. This would give a result of how the prototype performed in showing the relevant article and how the prototype performed in missing relevant articles.

6. Conclusions

To conclude the results, all the LLM techniques performed well and would be a good alternative to assessing if an article is relevant to Bosch. The multi-stage technique performed best in finding lower scored and less relevant articles, and few-shot performed best in finding the higher scored and more relevant articles. When using an LLM in a prototype, the RAG methods would be a good choice because it could be trained over time with what the user thinks is relevant.

This chapter will present a conclusion about the results by answering the problem description questions, discuss ethical aspects, and some future improvements.

6.1 Problem description questions

This section goes back to the questions posed in the problem description.

1. **How can an LLM be used to find relevant news articles for Bosch and be improved in its accuracy when assessing whether an article is relevant?**

To use an LLM to assess whether news articles are relevant to Bosch, an article could be sent in to the LLM along with a proper prompt, giving a "relevance score". The prompt to the LLM can be structured according to the zero-shot base prompt, as well as using different techniques.

An LLM can be improved in its accuracy when assessing an article's relevance by using a proper base prompt and by using the different LLM techniques that were tested in this report. The result shows that few-shot and the multi-stage pipeline are those two LLM techniques which improve accuracy the most, but the few-shot result can be questioned. The RAG is the best technique to find relevant articles, and the multi-stage pipeline is the best technique to find irrelevant articles but sometimes filtering out relevant articles. But because it is more important for Bosch to find the relevant articles, the RAG will be the best technique to assess if the article is relevant to Bosch.

2. **What requirements does a web-based prototype for information gathering need to fulfill to improve information scouting at Bosch?**

From an interview with a Bosch employee, many requirements were found based on their wishes and the shortcomings of the previous tools used. These requirements were compiled into a requirement specification. Broadly speaking, the most important requirements were that the prototype should find news articles on the internet related to a search query, or from specific websites and only the articles that are highly relevant and interesting to a Bosch employee should be presented to the end user. Other requirements were also found, all of which can be found in Appendix A.

3. **How can information scouting using an LLM be integrated into a web-based prototype?**

It was integrated into a web-based prototype by allowing a user to create several research projects, where inside each project articles would be fetched from a SERP API, and from given news source URLs. The articles found were then evaluated by an LLM using RAG, and only articles deemed relevant were presented to the user. On top of this, an LLM was used to summarize news articles for each project and all projects, as well as recommending news sources based on a search query.

4. **How well does the prototype meet Bosch's needs?**

From the user tests, a conclusion can be made that the prototype meets Bosch's needs since most of the test feedback was positive, and most of the initial requirements were met. While there is room for further development and improvements, the prototype is in such a state that it can be used for information scouting by an employee in their daily work.

6.2 Ethical aspects

The ethical aspects of LLM are highly topical today, and a large discussion is about copyright of the data in LLMs and the environmental cost of creating and using LLMs.

6.2.1 Copyright

LLM's are built upon a large number of training data, and a highly debated topic is the problem of what data are used in training and how this could harm the creators of data. The AI technologies are evolving fast which requires that the regulation about it also needs to evolve and currently there is no law that says the LLMs must disclose exactly what data has been used for training. This means that some LLMs can have been trained on unauthorized data, which can be seen as a copyright issue.

OpenAI claims that their models only are trained on publicly available data and licensed data. But they are not transparent with what data sources they are using when training their model which means that they maybe can have been used unauthorized data. Some studies show an indication that the model probably is trained on unauthorized data.

OpenAI also claims that they are following the fair use restriction which means that there is a limited usage of unauthorized data. OpenAI means that they don't store the data, only use it to train the model and that the generation of texts that LLMs does are new text which means that legally they don't need permission for the data they use because they never saves the data, they only trains a model, and they do not copy any of the maybe unauthorized data.

Since this is a grey zone of whether they are using unauthorized data or not, it has become a controversial topic. The problem is that if the LLM is trained on unauthorized data, it will have knowledge about the data and can generate text similar to the original and unauthorized data. It is hard to say if this is copyright or not because the original data is never stored and never used in the output. But it can be seen as unethical if they are using unauthorized data because the creators don't get any credits for doing their work, even though the LLM are trained or their data and can generate similar text as the creator. Because this is a big problem, there is a solution to new law regulation that will prevent companies from using unauthorized data for training their LLM models. [58][59][60][61][62]

6.2.2 Enviromental cost

To create and train an LLM has a negative effect on the environment in both the aspect of water usage and carbon emission. The scale of environmental effect depends on how many parameters the LLM has and how much training data is used. OpenAI's GPT-4 has 1.7 trillion parameters and consumes about 7200 MWh of electricity, and OpenAI's GPT-3 consumes about 700 000 liters of fresh water to be trained and about 500 ml of fresh water for every 10-50 answers it generates. [65] When using the LLM on a larger scale, the water and electricity

consumption will increase and will have a negative effect on the environment. This would not be sustainable over time and is an aspect to think about when using LLM's in a product.

6.2.3 Job opportunities

Since LLM's are performing well in logical thinking tasks and can compete with humans in logical tasks, it will be a possible alternative to human employees. LLM's are a much cheaper alternative and are a better candidate to use than hiring a human to a company. The LLMs still have some weakness in hallucinating in their answers, which does not make them as reliable as a human employee. But in the long term, if LLMs are improving, it could be a good alternative to human employees and would probably conquer the job market.

6.3 Future improvements

Individual user tests
Assessing whether an article is relevant or not relevant could depend on subjective opinions. This report tries to find a general method for the entire company Bosch to give out relevance scores, but because relevance is a subjective opinion, research whether the LLM techniques works better if the prompt is even more adapted to the user and not to a general instruction could improve the result of the relevance evaluation. This future improvement would also require further research into how the web application should be designed to be adapted for the user.

Combining different techniques
In table 4.5 (binary evaluation), it shows that multi-stage pipeline technique performed better in finding the negative articles while few-shots performed better in finding positive articles. A future improvement could be to combine these two techniques where in the second stage in multi-stage pipeline could use the few-shot technique, so the multi-stage pipeline becomes better at finding the positive articles.

Fine tuning
An LLM is pretrained on a large amount of data and is created to have general knowledge which can make it perform worse in more specific tasks. Fine-tuning is a method that trains the LLM and updates the parameters in the model, so it gets more domain-specific expertise. The training is a supervised learning method and involves having a dataset of instructions and output pairs. A future improvement of the research could be to test if fine tuning can improve the result of evaluating articles. With an LLM that has more domain-specific expertise, this could improve the results. [63]

To use fine-tuning, it requires different amounts of instruction and output pairs to see an improved result, but in general it requires about 50 sets of pairs. [64]

References

[1] OpenAI, "GPT-4o mini: Advancing cost-efficient intelligence," openai.com. Accessed: May 11, 2025. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[2] Stryker, Cole. “What is AI?”. ibm.com. Accessed: Jan. 21, 2025. [Online]. Available: https://www.ibm.com/think/topics/artificial-intelligence

[3] “What are large language models (LLMs)?”. ibm.com. Accessed: March 26, 2025. [Online]. Available: https://www.ibm.com/think/topics/large-language-models

[4] Microsoft, "What Is Deep Learning?". azure.microsoft.com. Accessed: March 26, 2025. [Online] Available: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-deep-learning/

[5] IBM. “What is a Neural Network?” ibm.com. Accessed March 26, 2025. [Online]. Available: https://www.ibm.com/think/topics/neural-networks

[6] Microsoft. “What Are Large Language Models (LLMs)?” azure.microsoft.com. Accessed March 26, 2025. [Online]. Available: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-large-language-models-llms

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention Is All You Need,” 2017, arXiv:1706.03762.

[8] OpenAI (OpenAI, San Francisco, CA, USA). “GPT-4o System Card” (2024). Accessed: March 26, 2025. [Online]. Available: https://cdn.openai.com/gpt-4o-system-card.pdf

[9] A. Ben Abacha et al. “MEDEC: A Benchmark for Medical Error Detection and Correction in Clinical Notes,” 2024, arXiv:2412.19260v1.

[10] DocsBot AI. “GPT-4o vs GPT-4o Mini: Detailed Performance & Feature Comparison”. Docsbot.ai. Accessed: May 11, 2025. [Online]. Available: https://docsbot.ai/models/compare/gpt-4o-mini/gpt-4o

[11] J. D. Workum et al. “Comparative evaluation and performance of large language models on expert level critical care questions: a benchmark study,” Critical Care, London, UK, vol. 29, no. 72, Feb. 2025. Accessed: March 26, 2025. [Online]. Available: https://link.springer.com/article/10.1186/s13054-025-05302-0

[12] Microsoft. “Azure OpenAI Service REST API Reference.” learn.microsoft.com. Accessed May 11, 2025. [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/openai/reference

[13] P. Thomas et al. (2024). “Large Language Models can Accurately Predict Searcher Preferences”. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’24). [Online]. Available: https://doi.org/10.1145/3626772.3657707

[14] J. A. Schnabel, J. R. Trippas, F. Scholer, and D. Hettiachchi, “Multi-stage Large Language Model Pipelines Can Outperform GPT-4o in Relevance Assessment,” 2025.

arXiv:2501.14296 Multi-stage Large Language Model Pipelines Can Outperform GPT-4o in Relevance Assessment

[15] M. Alaofi et al. (2024) “LLMs can be Fooled into Labelling a Document as Relevant”. In Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region. [Online]. Available: https://dl.acm.org/doi/10.1145/3673791.3698431

[16] L. Joos, D.A. Keim, M. T. Fischer. “Cutting Through the Clutter: The Potential of LLMs for Efficient Filtration in Systematic Literature Reviews”. 2025. ArXiv:2501.14296 Cutting Through the Clutter: The Potential of LLMs for Efficient Filtration in Systematic Literature Reviews

[17] G. Faggioli et al. “Perspectives on Large Language Models for Relevance Judgment”. 2025. ArXiv: 2304.09161

[18] J. White et al. “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT”. 2023. ArXiv: 2302.1138

[19] Microsoft. “Zero-shot and few-shot learning”. learn.microsoft.com. Accessed: February 11, 2025. [Online]. Available at: https://learn.microsoft.com/en-us/dotnet/ai/conceptual/zero-shot-learning

[20] B. Chen et al. “Unleasing the potential of prompt engineering in Large Language Models: a comprehensive review”. 2024. ArXiv: 2310.14735

[21] P. Liang et al. “Holistic Evaluation of Language Models”. 2023. ArXiv: 2211.09110.

[22] L. Reynolds and K. McDonell. (2021) “Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm”. In CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI ’21 Extended Abstracts). [Online]. Available at: https://dl.acm.org/doi/pdf/10.1145/3411763.3451760?casa_token=gNN8C9ceCTwAAAAA:AujhM5_vBvVBYDNsjX2rNsiToPgbxP5ge8S7pXRIlEG1RzH2ljSRUD__1XGhZj

[23] J. Wei et al. (2022). “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”. In 36th Conference on Neural Information Processing Systems (NeurIPS 2022). [Online]. Available at: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[24] Y. Gao et al. “Retrieval-Augmented Generation for Large Language Models: A survey”. 2024. ArXiv: 2312.10997 [2312.10997] Retrieval-Augmented Generation for Large Language Models: A Survey

[25] Z. Jing, Y. Su, Y. Han. “When Large Language Models Meet Vector Databases: A survey”. 2024. ArXiv: 2402.01763 2402.017

[26] Nilesh B. Korade, Mahendra B. Salunke, Amol A. Bhosle, Prashant B. Kumbharkar, Gayatri G. Asalkar and Rutuja G. Khedkar, “Strengthening Sentence Similarity Identification Through OpenAI Embeddings and Deep Learning”. International Journal of Advanced

Computer Science and Applications (IJACSA), vol 15. no 4, 2024. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2024.0150485

[27] OpenAI. “New and improved embedding model”. openai.com. Accessed: April 8, 2025. [Online]. Available: https://openai.com/index/new-and-improved-embedding-model/?utm_source=chatgpt.com

[28] N. Wiratunga et al. “CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question Answering”. 2024. ArXiv: 2404.04302.

[29] M. Goodwin. “What is an API?”. ibm.com. Accessed March 26, 2025. [Online]. Available: https://www.ibm.com/think/topics/api

[30] Microsoft. “Azure OpenAI Service.” azure.microsoft.com. Accessed March 26, 2025. [Online]. Available: https://azure.microsoft.com/en-us/products/ai-services/openai-service#Use-cases

[31] Microsoft. “What is Azure?”. azure.microsoft.com. Accessed March 26, 2025. [Online]. Available: What is Azure? | Microsoft Azure

[32] OpenAI. “OpenAI API.” openai.com. Accessed March 26, 2025. [Online]. Available: https://openai.com/api/

[33] S. Susnjara, I. Smalley. “Cloud Computing.” ibm.com. Accessed March 26, 2025. [Online]. Available: https://www.ibm.com/think/topics/cloud-computing

[34] Python Software Foundation. “About Python.” python.org. Accessed May 12, 2025. [Online]. Available: https://www.python.org/about/

[35] Anaconda, Inc. “Anaconda: The World’s Most Popular Data Science Platform.” anaconda.com. Accessed May 11, 2025. [Online]. Available: https://www.anaconda.com/

[36] FastAPI. “FastAPI: The Fastest Way to Build APIs with Python 3.7+.” fastapi.tiangolo.com. Accessed May 11, 2025. [Online]. Available: https://fastapi.tiangolo.com/

[37] Python Poetry. “Poetry - Python Dependency Management and Packaging Made Easy.” python-poetry.org. Accessed May 11, 2025. [Online]. Available: https://python-poetry.org/

[38] NumPy. “NumPy: The Fundamental Package for Scientific Computing with Python.” numpy.org. Accessed May 11, 2025. [Online]. Available: https://numpy.org/

[39] TypeScript. “TypeScript: JavaScript with Syntax for Types.” typescriptlang.org. Accessed May 11, 2025. [Online]. Available: https://www.typescriptlang.org/

[40] React. “React – A JavaScript Library for Building User Interfaces.” react.dev. Accessed May 11, 2025. [Online]. Available: https://react.dev/

[41] Bootstrap. “Bootstrap: The Most Popular HTML, CSS, and JavaScript Framework for Developing Responsive, Mobile-First Projects on the Web.” getbootstrap.com. Accessed May 11, 2025. [Online]. Available: https://getbootstrap.com/

[42] Microsoft. “What Is SQL Server?” learn.microsoft.com. Accessed May 11, 2025. [Online]. Available: https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16

[43] Qdrant Team. “Qdrant Documentation.” qdrant.tech. Accessed May 11, 2025. [Online]. Available: https://qdrant.tech/documentation/

[44] Google. “Google Scholar.” scholar.google.com. Accessed May 11, 2025. [Online]. Available: https://scholar.google.com/

[45] Lund University. “LUBsearch.” lub.lu.se. Accessed May 11, 2025. [Online]. Available: https://www.lub.lu.se/en/find/lubsearch

[46] A. Lantz, Intervjumetodik, 4th ed. Lund, Sweden: Studentlitteratur, 2013.

[47] N. Craswell et al. “OVERVIEW OF THE TREC 2019 DEEP LEARNING TRACK”. 2020. ArXiv: 2003.07820

[48] Scikit-learn. “Scikit-learn: Machine Learning in Python”. scikit-learn.org. Accessed May 11, 2025. [Online]. Available: https://scikit-learn.org/stable/index.html

[49] M. L. McHugh, “Interrater reliability: the kappa statistic”. Biochem. Med. (Zagreb), vol. 22, no. 3, pp. 276–282, 2012. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3900052/

[50] M. Li, Q. Gao, and T. Yu, “Kappa statistic considerations in evaluating inter-rater reliability between two raters: which, when and context matters”. BMC Cancer, vol. 23. Aug. 2023. [Online]. Available: https://bmccancer.biomedcentral.com/articles/10.1186/s12885-023-11325-z

[51] A. Doewes, N. A. Kurdhi, and A. Saxena. “Evaluating quadratic weighted kappa as the standard performance metric for automated essay scoring”. In Proceedings of the 16th International Conference on Educational Data Mining, pages 103–113, Bengaluru, India. 2023. International Educational Data Mining Society. [Online]. Available: https://pure.tue.nl/ws/portalfiles/portal/305425610/PUBLISHED_2023.EDM-long-papers.9.pdf

[52] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” Biometrics, vol. 33, no. 1, pp. 159–174, Mar. 1977. [Online]. Available: https://www.jstor.org/stable/2529310

[53] M. Pakula. “Using Small LLMs to Assess and Enhance Skill Management Documents”. M.S. thesis, Dept. of Computer Science, Lund University, Lund, Sweden, 2024. [Online]. Available: https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9175553&fileOId=9175554

[54] IBM, “What is three-tier architecture?”. IBM, Accessed May 11, 2025. [Online]. Available: https://www.ibm.com/think/topics/three-tier-architecture

[55] Serper, “Serper – The World's Fastest and Cheapest Google Search API”. serper.dev. Accessed May 8, 2025. [Online]. Available: https://serper.dev/

[56] L. Richardson, “Beautiful Soup Documentation”. readthedocs.io. Accessed May 8, 2025. [Online]. Available: https://beautiful-soup-4.readthedocs.io/en/latest/

[57] GeeksforGeeks, “User Acceptance Testing (UAT) – Software Testing”. geeksforgeeks.com. Apr. 2, 2025. [Online]. Available: https://www.geeksforgeeks.org/user-acceptance-testing-uat/

[58] M. Meeus et al. “Copyright Traps for Large Language Models.” arXiv, vol. 2402.09363, 2024. [Online]. Available: https://arxiv.org/pdf/2402.09363

[59] S. Rosenblat, T. O’Reilly, and I. Strauss, “Beyond public access in LLM pre-training data: Non-public book content in OpenAI’s models.” SSRC AI Disclosures Project Working Paper Series, no. 1, Social Science Research Council, Apr. 2025. [Online]. Available: https://ssrc-static.s3.us-east-1.amazonaws.com/OpenAI-Training-Violations-OReillyBooks_Sruly-OReilly-Strauss_SSRC_04012025.pdf

[60] J. Vincent, “Reported EU legislation to disclose AI training data could trigger copyright lawsuits.” theverge.com. Accessed: May 6, 2025 [Online]. Available: https://www.theverge.com/2023/4/28/23702437/eu-ai-act-disclose-copyright-training-data-report

[61] OpenAI, “Enterprise privacy at OpenAI.” openai.com. Accessed May 6, 2025. [Online]. Available: https://openai.com/enterprise-privacy/

[62] OpenAI, “OpenAI and journalism.” openai.com. Accessed May 6, 2025. [Online]. Available: https://openai.com/index/openai-and-journalism/

[63] S. Zhang et al. “Instruction tuning for large language models: A survey.” 2024. arXiv:2308.10792. [Online]. Available: https://arxiv.org/pdf/2308.10792

[64] OpenAI, “Fine-tuning – OpenAI API-” platform.openai.com. Accessed May 6, 2025. [Online]. Available: https://platform.openai.com/docs/guides/fine-tuning/when-to-use-fine-tuning

[65] O. Håkans. “The environmental impact of using Artificial Intelligence.” B.S. thesis. School of Electrical Eng. and Comp. Sci., KTH Royal Institute of technology., Stockholm, Sweden, 2025. [Online]. Available: https://kth.diva-portal.org/smash/get/diva2%3A1948879/FULLTEXT02.pdf

Appendix A - Requirement specification

Introduction

This document specifies the requirements for the prototype to be developed during our thesis. Our vision of the prototype is that it could replace the tool that Bosch previously used to scout for information on the internet. Therefore, the requirements are formulated and prioritized with this goal in mind. This is not to say that the prototype will improve on all the aspects of the previous tool, just in a few key areas that users deemed important. The relevance to the thesis was also taken into consideration.

Functional requirements are requirements that regard functionalities that can be implemented in code and can be verified in an unambiguous way. These are labelled FR. Non-functional requirements are less concrete or related to UX. They are labelled NFR.

Priority scale

The requirements were prioritized on a number scale from 1 to 3, where:

1- High priority. These requirements are necessary to fulfill to consider the prototype useful.
2- Medium priority. These requirements are functionalities that could improve the prototype but are not crucial for the users or thesis.
3- Low priority. These requirements are mostly nice to have and should be implemented last if there is time.

Requirements

Functional requirements

1 - High priority

FR1.1 The prototype should find news articles on the internet related to a search query.

FR1.2 Found articles should be evaluated by an LLM according to their relevance.

FR1.3 The prototype should present news articles to the user in a list, with each entry containing the following information: Title, short excerpt or summary, publication date, publisher/source, link to article.

FR1.4 The articles should be in English or Swedish.

FR1.5 The prototype should not present scientific publications to the user.

FR1.6 The user should be able to create several search queries that are updated with new articles on a regular basis.

FR1.7 The user should be able to filter out certain domains and websites.

FR1.8 The user should be able to save articles.

2 - Medium Priority

FR2.1 The user should be able to put a search query into “projects” that contain several search queries.

FR2.2 The user should be able to filter articles by publication date.

FR2.3 The articles should be in languages other than Swedish or English.

FR2.4 The articles should be able to be sorted according to date or relevance

FR2.5 The user should be able to label articles with a thumbs up/down to indicate whether they found it relevant.

FR2.6 Each article should contain an AI-generated relevance score that tells the user how relevant the article is.

FR2.7 An LLM-generated summary of most relevant found articles should be able to be presented to the user, both for each individual query, all queries, or each project's queries.

FR2.8 Each article should be tagged according to what areas in concerns.

FR2.9 The prototype should fetch articles from specific websites.

3 - Low priority
FR3.1 The prototype should, on top of news articles, also find other content on the internet.

FR3.2 The prototype should alert the user when new articles for a certain search category appear.

FR3.3 Occasionally, an article deemed less relevant should be presented to the user, and they should be prompted to mark it.

Non-functional requirements

1- High priority
NFR1.1 The articles presented should be relevant to the user and related to the search query. Articles that are related to the query but not relevant to the user should be filtered out.
NFR1.2 The articles presented should be related to new technologies, technological advancements, or new tech companies. Articles regarding what Bosch’s competitors are doing are also of interest. Other non-tech matters such as stock prices or organizational changes are of lower priority.
NFR1.3 For each article, a large part of the screen should be an excerpt from the article text, and a small part should be other information.
[paste design sketches here]

2- Medium Priority
NFR2.1 Previous articles marked with a “thumbs up” should be used by an LLM to assess relevance to newly found articles.

3- Low priority
FR3.1 An LLM should take the users search query input and rework it to generate a better search query that will increase the chance to get more relevant news articles.

Appendix B - Prompts

Zero-shot prompt

```
#Context
Assume that a user is looking for interesting news articles to find technological
developments and identify technical trends for the company Bosch.
The one searching is an employee at Bosch, whose job is to scout for new
technologies, technological advancements, or tech trends in areas and industries
that Bosch operate in.

# Task
Given a news article, you must decide if it is relevant for the company Bosch by
providing a score on an integer scale of 0 to 3 with the following meanings:
3 = Perfectly relevant: The article provides highly valuable information that
aligns directly with Bosch's technological advancements, strategic direction, or
areas of interest. It provides critical insights that would be of immediate
interest to the employee. Only award a score of 3 if the article presents highly
actionable, impactful, and specific insights that align directly with Bosch's
current technological focus or strategic goals.
2 = Highly relevant: The article provides significant value to Bosch but is not
as immediately actionable as a score of 3. It still addresses important
technological trends or competitive insights within Bosch’s industries. The
article is still highly useful but may not be immediately actionable. It
discusses important technological trends, major industry shifts, or competitor
activities that are relevant to Bosch but not groundbreaking.
1 = Relevant: The article has some relevance, such as mentioning technological
trends or innovations, but it’s not a direct fit with Bosch’s current strategic
or technological focus. It may be of moderate interest, but likely doesn’t
directly impact Bosch's immediate priorities. The article relates to Bosch's
industries or technologies. It might discuss trends, emerging technologies, or
general industry insights that are worth noting but provides less clear,
actionable intelligence.
0 = Not relevant: The article is not as relevant. It may be non-technical,
unrelated to Bosch’s areas of operation, or not a news article at all. The
content does not provide any actionable insights that would be of interest to a
Bosch employee. Articles that are tangentially relevant or discuss trends that
are only vaguely connected to Bosch’s business should receive a score of 0. Also
articles that are unrelated to Bosch’s technological and business focus, or for
content that is non-technical, untrustworthy, or irrelevant.

        #Examples of interesting topics and content
        Below are examples of article topics and content that would be relevant,
interesting and useful for the employee and Bosch:
        1. Companies or organizations making significant technological
advancements in technologies or industries that Bosch is involved in.
        2. A company or competitor releasing a new product that could compete
with Bosch.
        3. Companies or organizations investing in technologies or industries
relevant to Bosch.
        4. Companies investing in or acquiring other companies that are in the
same industries as Bosch.
        5. Collaborations or partnerships between companies that operate in
Bosch's industries.
```

```
        6. Roadmaps or strategy plans from companies or organization that are
relevant to Bosch’s industry or technological focus.
        7. Companies or products winning innovation awards or prizes related to
Bosch’s technological interests.

        # Article to be evaluated
        (The following is part of the HTML content of the website. Only take into
consideration what is actually part of the articles content, and ignore other
irrelevant text on the page.)
        Consider the following news article:
        --BEGIN ARTICLE CONTENT--
        {article_content}
        --END ARTICLE CONTENT--

        # Instructions
        When evaluating an article, consider the following steps:
        - Does the content align with Bosch's technological priorities and goals?
Does it discuss technologies, industries, or competitors directly related to
Bosch’s business?
        - Is the article about a significant, cutting-edge technology, trend, or
innovation that could impact Bosch's operations? Only advancements that are
clearly actionable or strategic for Bosch should score highly.
        - Is the article about a company or organization within the same industry
or field as Bosch? Does it mention companies that directly compete with or
complement Bosch’s technology?
        - Does the article provide actionable insights or valuable intelligence?
Is it something that could directly impact Bosch’s decision-making process,
technological strategy, or competitive standing?
        - Can the article be categorized into one of the topics previously
presented?
        - Does the article come from a reputable, reliable source? Prioritize
content from credible sources that are known for reporting on technological
trends and industry advancements.

        Consider the aspects above and the relative importance of each, evaluate
the overall relevance and decide on a final score. Be pessimistic when
evaluating, that is opt for a lower score if unsure.

        # Output format
        Summarize the most important take-aways of the article. Reason and
explain your score according to the criteria above, and produce an integer 0-3.
Respond in the following format, with the following fields:
        "summary": "Summary of the article, highlighting the most important take-
aways",
        "explanation": "Reasoning and explanation of the given score" ,
        "score": X
```

Few-shot prompt

#Context

Assume that a user is looking for interesting news articles to find technological developments and identify technical trends for the company Bosch.

The one searching is an employee at Bosch, whose job is to scout for new technologies, technological advancements, or tech trends in areas and industries that Bosch operate in.

Task

Given a news article, you must decide if it is relevant for the company Bosch by providing a score on an integer scale of 0 to 3 with the following meanings:

3 = Perfectly relevant: The article provides highly valuable information that aligns directly with Bosch's technological advancements, strategic direction, or areas of interest. It provides critical insights that would be of immediate interest to the employee. Only award a score of 3 if the article presents highly actionable, impactful, and specific insights that align directly with Bosch's current technological focus or strategic goals.

2 = Highly relevant: The article provides significant value to Bosch but is not as immediately actionable as a score of 3. It still addresses important technological trends or competitive insights within Bosch's industries. The article is still highly useful but may not be immediately actionable. It discusses important technological trends, major industry shifts, or competitor activities that are relevant to Bosch but not groundbreaking.

1 = Relevant: The article has some relevance, such as mentioning technological trends or innovations, but it's not a direct fit with Bosch's current strategic or technological focus. It may be of moderate interest, but likely doesn't directly impact Bosch's immediate priorities. The article relates to Bosch's industries or technologies. It might discuss trends, emerging technologies, or general industry insights that are worth noting but provides less clear, actionable intelligence.

0 = Not relevant: The article is not as relevant. It may be non-technical, unrelated to Bosch's areas of operation, or not a news article at all. The content does not provide any actionable insights that would be of interest to a Bosch employee. Articles that are tangentially relevant or discuss trends that are only vaguely connected to Bosch's business should receive a score of 0. Also articles that are unrelated to Bosch's technological and business focus, or for content that is non-technical, untrustworthy, or irrelevant.

#Examples of interesting topics and content

Below are examples of article topics and content that would be relevant, interesting and useful for the employee and Bosch:

1. Companies or organizations making significant technological advancements in technologies or industries that Bosch is involved in.
2. A company or competitor releasing a new product that could compete with Bosch.
3. Companies or organizations investing in technologies or industries relevant to Bosch.
4. Companies investing in or acquiring other companies that are in the same industries as Bosch.
5. Collaborations or partnerships between companies that operate in Bosch's industries.
6. Roadmaps or strategy plans from companies or organization that are relevant to Bosch's industry or technological focus.
7. Companies or products winning innovation awards or prizes related to Bosch's technological interests.

#Examples of news articles

Here is a list of a few previously evaluated articles and their given scores. Try to understand why each article got that given score, and use it when reasoning about future articles you evaluate.

(The given is part of the HTML content of the website. Only take into consideration what is actually part of the content, and what is other irrelevant text on the page.)

This article was given a score of 3.
--BEGIN SCORE 3 EXAMPLE ARTICLE--
{example_3}
--END SCORE 3 EXAMPLE ARTICLE--

This article was given a score of 2.
--BEGIN SCORE 2 EXAMPLE ARTICLE--
{example_2}
--END SCORE 2 EXAMPLE ARTICLE--

This article was given a score of 1.
--BEGIN SCORE 1 EXAMPLE ARTICLE--
{example_1}
--END SCORE 1 EXAMPLE ARTICLE--

This article was given a score of 0.
--BEGIN SCORE 0 EXAMPLE ARTICLE--
{example_0}
--END SCORE 0 EXAMPLE ARTICLE--

Article to be evaluated
(The following is part of the HTML content of the website. Only take into consideration what is actually part of the articles content, and ignore other irrelevant text on the page.)
Consider the following news article:
--BEGIN ARTICLE CONTENT--
{article_content}
--END ARTICLE CONTENT--

Instructions
When evaluating an article, consider the following steps:
- Does the content align with Bosch's technological priorities and goals? Does it discuss technologies, industries, or competitors directly related to Bosch’s business?
- Is the article about a significant, cutting-edge technology, trend, or innovation that could impact Bosch's operations? Only advancements that are clearly actionable or strategic for Bosch should score highly.
- Is the article about a company or organization within the same industry or field as Bosch? Does it mention companies that directly compete with or complement Bosch’s technology?
- Does the article provide actionable insights or valuable intelligence? Is it something that could directly impact Bosch’s decision-making process, technological strategy, or competitive standing?
- Can the article be categorized into one of the topics previously presented?
- Does the article come from a reputable, reliable source? Prioritize content from credible sources that are known for reporting on technological trends and industry advancements.

Consider the aspects above and the relative importance of each, evaluate the overall relevance and decide on a final score. Be pessimistic when evaluating, that is opt for a lower score if unsure.

Output format
Summarize the most important take-aways of the article. Reason and explain your score according to the criteria above, and produce an integer 0-3. Respond in the following format, with the following fields:
"explanation": "Reasoning and explanation of the given score" ,
"score": X

Multi-stage pipeline prompt

Stage 1

#Context

Assume that a user is looking for interesting news articles to find technological developments and identify technical trends for the company Bosch.

The one searching is an employee at Bosch, whose job is to scout for new technologies, technological advancements, or tech trends in areas and industries that Bosch operate in.

This employee more specifically works with and is interested in connectivity, connected vehicles, V2X, cloud -and edge computing, cyber security, and post-quantum cryptography.

Task

1 = Relevant: The article has at least some relevance, such as mentioning technological trends or innovations. It may be of moderate interest or higher. The article relates to Bosch's industries or technologies. It might discuss trends, emerging technologies, or general insights.

0 = Not relevant: The article is not as relevant. It may be non-technical, unrelated to Bosch's areas of operation, or not a news article at all. The content does not provide any actionable insights that would be of interest to a Bosch employee. Articles that are tangentially relevant or discuss trends that are only vaguely connected to Bosch's business should receive a score of 0. Also articles that are unrelated to Bosch's technological and business focus, or for content that is non-technical, untrustworthy, or irrelevant.

#Examples of interesting topics and content

Below are examples of article topics and content that would be relevant, interesting and useful for the employee and Bosch:

1. Companies or organizations making significant technological advancements in technologies or industries that Bosch is involved in.
2. A company or competitor releasing a new product that could compete with Bosch.
3. Companies or organizations investing in technologies or industries relevant to Bosch.
4. Companies investing in or acquiring other companies that are in the same industries as Bosch.
5. Collaborations or partnerships between companies that operate in Bosch's industries.
6. Roadmaps or strategy plans from companies or organization that are relevant to Bosch's industry or technological focus.
7. Companies or products winning innovation awards or prizes related to Bosch's technological interests.

###Article

Consider the following news article:
--BEGIN ARTICLE CONTENT--
{article_content}
--END ARTICLE CONTENT--

Instructions

Lets think step by step:

- Does the content align with Bosch's technological priorities and goals? Does it discuss technologies, industries, or competitors directly related to Bosch's business?
- Is the article about a significant, cutting-edge technology, trend, or innovation that could impact Bosch's operations? Only advancements that are clearly actionable or strategic for Bosch should score highly.

- Is the article about a company or organization within the same industry or field as Bosch? Does it mention companies that directly compete with or complement Bosch's technology?
- Does the article provide actionable insights or valuable intelligence? Is it something that could directly impact Bosch's decision-making process, technological strategy, or competitive standing?
- Can the article be categorized into one of the topics previously presented?
- Does the article come from a reputable, reliable source? Prioritize content from credible sources that are known for reporting on technological trends and industry advancements.

Consider the aspects above and the relative importance of each, evaluate the overall relevance and decide on a final score. Be pessimistic when evaluating, that is give a lower score if unsure.

Output format
Only produce an integer 0 or 1. Output must be an integer value only.

Stage 2
Same as xero-shot prompt

RAG 1 prompt

#Context Assume that a user is looking for interesting news articles to find technological developments and identify technical trends for the company Bosch. The one searching is an employee at Bosch, whose job is to scout for new technologies, technological advancements, or tech trends in areas and industries that Bosch operate in. This employee more specifically works with and is interested in connectivity, connected vehicles, V2X, cloud -and edge computing, cyber security, and post-quantum cryptography.

Task

Given a news article, you must decide if it is relevant for the company Bosch by providing a score on an integer scale of 0 to 3 with the following meanings:

3 = Perfectly relevant: The article provides highly valuable information that aligns directly with Bosch's technological advancements, strategic direction, or areas of interest. It provides critical insights that would be of immediate interest to the employee. Only award a score of 3 if the article presents highly actionable, impactful, and specific insights that align directly with Bosch's current technological focus or strategic goals.

2 = Highly relevant: The article provides significant value to Bosch but is not as immediately actionable as a score of 3. It still addresses important technological trends or competitive insights within Bosch's industries. The article is still highly useful but may not be immediately actionable. It discusses important technological trends, major industry shifts, or competitor activities that are relevant to Bosch but not groundbreaking.

1 = Relevant: The article has some relevance, such as mentioning technological trends or innovations, but it's not a direct fit with Bosch's current strategic or technological focus. It may be of moderate interest, but likely doesn't directly impact Bosch's immediate priorities. The article relates to Bosch's industries or technologies. It might discuss trends, emerging technologies, or general industry insights that are worth noting but provides less clear, actionable intelligence.

0 = Not relevant: The article is not as relevant. It may be non-technical, unrelated to Bosch's areas of operation, or not a news article at all. The content does not provide any actionable insights that would be of interest to a Bosch employee. Articles that are tangentially relevant or discuss trends that are only vaguely connected to Bosch's business should receive a score of 0. Also articles that are unrelated to Bosch's technological and

business focus, or for content that is non-technical, untrustworthy, or irrelevant.

#Examples of interesting topics and content
Below are examples of article topics and content that would be relevant, interesting and useful for the employee and Bosch:
1. Companies or organizations making significant technological advancements in technologies or industries that Bosch is involved in.
2. A company or competitor releasing a new product that could compete with Bosch.
3. Companies or organizations investing in technologies or industries relevant to Bosch.
4. Companies investing in or acquiring other companies that are in the same industries as Bosch.
5. Collaborations or partnerships between companies that operate in Bosch's industries.
6. Roadmaps or strategy plans from companies or organization that are relevant to Bosch's industry or technological focus.
7. Companies or products winning innovation awards or prizes related to Bosch's technological interests.

#Examples of news articles
Here is a list of a few previously evaluated articles and their given scores. Understand why each article got that given score, and use it when reasoning about future articles you evaluate.
(The given is part of the HTML content of the website. Only take into consideration what is actually part of the content, and what is other irrelevant text on the page.)

This article was given a score of {retrieved_0_score}.
--BEGIN SCORE {retrieved_0_score} EXAMPLE ARTICLE--
{retrieved_0}
--END SCORE {retrieved_0_score} EXAMPLE ARTICLE--

This article was given a score of {retrieved_1_score}.
--BEGIN SCORE {retrieved_1_score} EXAMPLE ARTICLE--
{retrieved_1}
--END SCORE {retrieved_1_score} EXAMPLE ARTICLE--

This article was given a score of {retrieved_2_score}.
--BEGIN SCORE {retrieved_2_score} EXAMPLE ARTICLE--
{retrieved_2}
--END SCORE {retrieved_2_score} EXAMPLE ARTICLE--

This article was given a score of {retrieved_3_score}.
--BEGIN SCORE {retrieved_3_score} EXAMPLE ARTICLE--
{retrieved_3}
--END SCORE {retrieved_3_score} EXAMPLE ARTICLE--

Article to be evaluated
(The following is part of the HTML content of the website. Only take into consideration what is actually part of the articles content, and ignore other irrelevant text on the page.)
Consider the following news article:
--BEGIN ARTICLE CONTENT--
{article_content}
--END ARTICLE CONTENT-

Instructions

When evaluating an article, consider the following steps:
- Does the content align with Bosch's technological priorities and goals? Does it discuss technologies, industries, or competitors directly related to Bosch's business?
- Is the article about a significant, cutting-edge technology, trend, or innovation that could impact Bosch's operations? Only advancements that are clearly actionable or strategic for Bosch should score highly.
- Is the article about a company or organization within the same industry or field as Bosch? Does it mention companies that directly compete with or complement Bosch's technology?
- Does the article provide actionable insights or valuable intelligence? Is it something that could directly impact Bosch's decision-making process, technological strategy, or competitive standing?
- Can the article be categorized into one of the topics previously presented?
- Does the article come from a reputable, reliable source? Prioritize content from credible sources that are known for reporting on technological trends and industry advancements.

Consider the aspects above and the relative importance of each, evaluate the overall relevance and decide on a final score. Be pessimistic when evaluating, that is opt for a lower score if unsure.

Output format
Summarize the most important take-aways of the article. Reason and explain your score according to the criteria above, and produce an integer 0-3. Respond in the following format, with the following fields:
"summary": "Summary of the article, highlighting the most important take-aways",
"explanation": "Reasoning and explanation of the given score" ,
"score": X

RAG 2 prompt

#Context
Assume that a user is looking for interesting news articles to find technological developments and identify technical trends for the company Bosch.
The one searching is an employee at Bosch, whose job is to scout for new technologies, technological advancements, or tech trends in areas and industries that Bosch operate in.
This employee more specifically works with and is interested in connectivity, connected vehicles, V2X, cloud -and edge computing, cyber security, and post-quantum cryptography.

Task
Given a news article, you must decide if it is relevant for the company Bosch by providing a score on an integer scale of 0 to 3 with the following meanings:
3 = Perfectly relevant: The article provides highly valuable information that aligns directly with Bosch's technological advancements, strategic direction, or areas of interest. It provides critical insights that would be of immediate interest to the employee. Only award a score of 3 if the article presents highly actionable, impactful, and specific insights that align directly with Bosch's current technological focus or strategic goals.
2 = Highly relevant: The article provides significant value to Bosch but is not as immediately actionable as a score of 3. It still addresses important technological trends or competitive insights within Bosch's industries. The article is still highly useful but may not be immediately actionable. It discusses important technological trends, major industry shifts, or competitor activities that are relevant to Bosch but not groundbreaking.
1 = Relevant: The article has some relevance, such as mentioning technological trends or innovations, but it's not a direct fit with Bosch's current strategic or technological focus. It

may be of moderate interest, but likely doesn't directly impact Bosch's immediate priorities. The article relates to Bosch's industries or technologies. It might discuss trends, emerging technologies, or general industry insights that are worth noting but provides less clear, actionable intelligence.

0 = Not relevant: The article is not as relevant. It may be non-technical, unrelated to Bosch's areas of operation, or not a news article at all. The content does not provide any actionable insights that would be of interest to a Bosch employee. Articles that are tangentially relevant or discuss trends that are only vaguely connected to Bosch's business should receive a score of 0. Also articles that are unrelated to Bosch's technological and business focus, or for content that is non-technical, untrustworthy, or irrelevant.

#Examples of interesting topics and content

Below are examples of article topics and content that would be relevant, interesting and useful for the employee and Bosch:

1. Companies or organizations making significant technological advancements in technologies or industries that Bosch is involved in.
2. A company or competitor releasing a new product that could compete with Bosch.
3. Companies or organizations investing in technologies or industries relevant to Bosch.
4. Companies investing in or acquiring other companies that are in the same industries as Bosch.
5. Collaborations or partnerships between companies that operate in Bosch's industries.
6. Roadmaps or strategy plans from companies or organization that are relevant to Bosch's industry or technological focus.
7. Companies or products winning innovation awards or prizes related to Bosch's technological interests.

#Examples of news articles

Here is a list of a few previously evaluated articles and their given scores. Understand why each article got that given score, and use it when reasoning about future articles you evaluate.

(The given is part of the HTML content of the website. Only take into consideration what is actually part of the content, and what is other irrelevant text on the page.)

This article was given a score of 3.
--BEGIN SCORE 3 EXAMPLE ARTICLE--
{example_3}
--END SCORE 3 EXAMPLE ARTICLE--

This article was given a score of 2.
--BEGIN SCORE 2 EXAMPLE ARTICLE--
{example_2}
--END SCORE 2 EXAMPLE ARTICLE--

This article was given a score of 1.
--BEGIN SCORE 1 EXAMPLE ARTICLE--
{example_1}
--END SCORE 1 EXAMPLE ARTICLE--

This article was given a score of 0.
--BEGIN SCORE 0 EXAMPLE ARTICLE--
{example_0}
--END SCORE 0 EXAMPLE ARTICLE--

Article to be evaluated

(The following is part of the HTML content of the website. Only take into consideration what is actually part of the articles content, and ignore other irrelevant text on the page.)

Consider the following news article:

--BEGIN ARTICLE CONTENT--

{article_content}

--END ARTICLE CONTENT--

Instructions

When evaluating an article, consider the following steps:

- Does the content align with Bosch's technological priorities and goals? Does it discuss technologies, industries, or competitors directly related to Bosch's business?
- Is the article about a significant, cutting-edge technology, trend, or innovation that could impact Bosch's operations? Only advancements that are clearly actionable or strategic for Bosch should score highly.
- Is the article about a company or organization within the same industry or field as Bosch? Does it mention companies that directly compete with or complement Bosch's technology?
- Does the article provide actionable insights or valuable intelligence? Is it something that could directly impact Bosch's decision-making process, technological strategy, or competitive standing?
- Can the article be categorized into one of the topics previously presented?
- Does the article come from a reputable, reliable source? Prioritize content from credible sources that are known for reporting on technological trends and industry advancements.

Consider the aspects above and the relative importance of each, evaluate the overall relevance and decide on a final score. Be pessimistic when evaluating, that is opt for a lower score if unsure.

Output format

Summarize the most important take-aways of the article. Reason and explain your score according to the criteria above, and produce an integer 0-3. Respond in the following format, with the following fields:

"summary": "Summary of the article, highlighting the most important take-aways",
"explanation": "Reasoning and explanation of the given score" ,
"score": X

Printed by Tryckeriet i E-huset, Lund 2025

LUND
UNIVERSITY

Series of Bachelor`s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2025-1064
http://www.eit.lth.se