WEST UNIVERSITY OF TIMIŞOARA
FACULTY OF COMPUTER SCIENCE
BACHELOR STUDY PROGRAM: COMPUTER SCIENCE IN ENGLISH

BACHELOR THESIS

SUPERVISOR:
Prof./Conf./Lect. Dr. Monica Sancira

GRADUATE:
Daniel-Mihai Muntean

TIMIŞOARA
2026

WEST UNIVERSITY OF TIMIŞOARA
FACULTY OF COMPUTER SCIENCE
BACHELOR STUDY PROGRAM: COMPUTER SCIENCE IN ENGLISH

Title

SUPERVISOR:
Prof./Conf./Lect. Dr. Monica Sancira

GRADUATE:
Daniel-Mihai Muntean

TIMIŞOARA
2026

Abstract

The rapid growth of online information presents both opportunities and challenges for data-driven applications. While vast amounts of data are publicly available, they are often distributed across multiple platforms with diverse formats, structures, and access restrictions, making collection and integration difficult. This thesis proposes the development of an application that combines web automation techniques with artificial intelligence to efficiently collect, unify, and interpret heterogeneous data.

The system integrates the use of public and private APIs for data collection and controlled bypassing of certain restrictions to gather information from various sources. Collected data are then processed using commercial AI APIs, enabling contextual understanding and conversion into a consistent, unified format. This approach allows flexible adaptation to different use cases, including advanced information retrieval, trend analysis, and automated summarization.

The main contribution of this work is a hybrid framework that bridges the gap between distributed data sources and intelligent applications. By providing a robust method for automated data collection and AI-driven interpretation, the thesis demonstrates a practical solution for extracting meaningful insights from diverse online information. The proposed approach also addresses challenges related to data quality, semantic consistency, and ethical considerations in automated data processing.

Statement of Scientific integrity

I declare that this BSc / MSc thesis is an original report and represents my own work, conducted under supervision of Prof./Conf./Lect. Dr. Monica Sancira. All research methods, data collection, analysis and interpretation presented in this thesis are accurately reported and all sources have been cited appropriately where I have made use of ideas and/or visuals of others.

If you did not use generative AI tools: I did not use generative AI during the writing and research process.

If you used generative AI tools: I used generative AI during the writing and research process. The details are in the Appendix A.

Contents

Chapter 1

Introduction

1.1 Addressed Problem

The architecture of digital information retrieval is currently undergoing its most significant transformation. Historically, search engines have been the main way people access information on the internet, mapping user queries to indexed web pages based on lexical relevance. However, contemporary data indicates a profound shift in consumer behavior driven by a mounting intolerance for the friction inherent in modern web navigation. Traditional browsing requires users to traverse a highly fragmented ecosystem characterized by aggressive advertising real estate, restrictive paywalls and heavily manipulated content that often obscures the targeted information.

This friction caused a mass migration toward AI conversational agents. Recent evaluations demonstrate that the traditional search engine market share, which consistently hovered above 90 percent for over a decade, fell bellow this threshold in late 2024 (`https://searchengineland.com/google-search-market-share-drops-2024-450497`), tied directly with the rapid adoption of large language models such as ChatGPT*, Claude*, Gemini*, etc. Approximately 80 percent of consumers now rely on AI-generated summaries for at least 40 percent of their digital queries. (`https://towcenter.columbia.edu/sites/towcenter.columbia.edu/files/content/Journalism%20Zero_%20How%20Platforms%20and%20Publishers%20are%20Navigating%20AI.pdf`)

Studies tracking web browsing behavior found that when an AI summary is present, the propensity of a user to click on a cited source link diminishes to near zero. (`https://www.pewresearch.org/wp-content/uploads/sites/20/2025/05/pl_2025.05.23_metered-data-ai_report.pdf`)

Despite this preference, the shift introduces severe new vulnerabilities regarding data accuracy. While users desire the frictionless interface of a chat bot, general purpose conversational models are probabilistically driven and inherently lack the grounding required for high stakes accuracy. The thesis under consideration addresses this precise intersection: it proposes a web-based application that leverages the preferred conversational synthesis interface while enforcing rigorous data accuracy through a platform-specific architectural constraint.

In applied settings, hallucinations typically originate from three primary sources within the LLM development pipeline:

Pre-training Data Limitations: General-purpose models are trained on mas-

sive, unfiltered data found on the internet. Neural networks possess the tendency to memorize this training data, consequently internalizing the noise and conflicting information present in the source material.

Parametric Knowledge Boundaries: A model's internal memory is static post-training. When queried on specialized domains, such as obscure cinematographic metadata or highly specific film pedigrees, the model frequently exhibits deficits in domain-specific knowledge. Instead of failing gracefully, the model attempts to interpolate across these knowledge gaps, resulting the fabrication of false facts.

Prompt-Driven: Complex or ambiguous user instructions can force the model into erratic neural pathways. Studies utilizing attribution frameworks demonstrate that LLMs frequently prioritize formatting compliance or conversational flow over factual fidelity when presented with tricky prompts. (`https://dr-arsanjani.medium.com/navigating-the-challenges-of-hallucinations-in-llm-applications-strategies-and`

To combat these inherent flaws, contemporary literature strongly advocates for "platform-specific AI implementation." While general models like Gemini or GPT-4 are highly versatile, they lack the specialized grounding required for deterministic accuracy. Advanced mitigation methods emphasize the necessity of separating the processing engine (LLM) from the factual knowledge base (`https://arxiv.org/html/2510.06265v1`). By pre-conditioning the model with structured, domain-specific data and restricting user inputs, drastically reduce the rate of factual fabrication.

1.2 Motivation

In the current information era, the amount and variety of data available on the internet are growing extremely fast. The real value of information depends not only on how much data can be collected, but also on how efficiently it can be integrated, interpreted, and used. However, relevant data is distributed across many different platforms, with inconsistent formats, access restrictions, and structures. This creates serious challenges for both research and practical applications. Many existing data collection systems are limited to specific sources and cannot ensure consistent results when combining heterogeneous data.

This thesis proposes the development of an application that automates data collection, filtering and analysis using commercial artificial intelligence APIs, converting data into a consistent, unified format for a more scalable analysis of target data from multiple sources. This approach allows the program to adapt to different use cases, such as smart information search, trend monitoring, or automatic text interpretation.

The main motivation for this work is the need for a flexible and intelligent solution that can reduce the gap between distributed data sources and the applications that rely on them. The proposed system aims to solve three main problems:

- the lack of tools that can combine multiple data collection methods,

- the difficulty of unifying data from heterogeneous sources, and
- the use of abstract prompts that fetch bad/broken and corrupts it through AI prompting, causing it to halucinate

Personally, this topic combines my interests in software development, artificial intelligence and data engineering. By developing and testing this system, the thesis aims to create a practical tool for intelligent data collection and to contribute to the understanding of hybrid architectures that connect web automation with AI interpretation.

1.3 Aim of Paper

This thesis proposes the development of a web-based application which should help users to retrieve information on the topic of cinematographic works implementing a data pipeline that organizes semi-structured movie metadata from external APIs in an easily accessible and digestible format with highly accurate information output.

By applying specialized prompt engineering methods that enable Large Language Models (LLMs) to perform the different types of analysis on the target, queried data extracted from large data sets and evaluating the effectiveness of AI driven synthesis of data compared to traditional manual metadata retrieval.

1.4 Methodology

The development of the proposed application follows a structured data driven pipeline, transitioning from raw data acquisition to refined semantic synthesis. The process is divided into the following phases: data retrieval, filtering of retrieved data, heuristic prompt engineering and user friendly presentation.

The initial and most important stage for this method is the data acquisition. Using the Kaggle API*, the system queries a comprehensive dataset containing the IMDb* database (it is regularly updated to contain even the latest titles) containing cinematographic titles, release years, their audience ratings and all their details which can be manually accessible on IMDb.

1.4.1 Data Acquisition and Preprocessing

- The system filters results based on the user's specific search query over the IMDb dataset.
- With taking into account the ease of use for the software, the user is shown titles that may be relevant to their search, leaving the choice of analysis selection to the user.
- With the selection made, the data for the respective title is extracted from the data set, filtering it for relevant information that would guarantee us the best results for our custom tailored prompt to the Large Language Model analysis step.

Semantic Analysis using Large Language Models

- The core of the methodology lies in the integration of the Gemini API* as a reasoning and interpreting engine. Rather than using a generic chat interface which would require the manual preparation of a prompt for the user in order to get their results, the system employs a targeted predefined template which includes all relevant information from the extracted metadata, ensuring the results (narrative summarization, pedigree evaluation, critical verdict, audience segmentation, stylistic recommendations) are of the upmost quality and accuracy.

- By providing the model with the exact data it needs to process, the AI is forced to stick to the provided context rather than wandering into unrelated neural pathways that may cause it to hallucinate and give us bad results.

Output Serialization and UI Integration

- The final phase focuses on the transition from raw text data, into a structured format suitable for an user friendly application interface.

- The LLM is instructed to output results using a restricted set of HTML tags which ensures that the unstructured text mining results are immediately presentable to users without requiring additional front end parsing.

- The final result provides the user with a "verdict" on the target topic, giving useful information which should clarify the users questions: "Is this worth watching?", "What other movies would be similar?"

Chapter 2

State of the art

Whilst platforms that provide specific information for certain topics are still very popular, the general user started shifting their preference to AI chat bots. Why? Because it eliminates the need of researching to find the simple answer to their probably simple question. No trying sites one after the other because of ads, paywalls or scams. This transition is driven by the user's desire to bypass the 'friction' of modern web navigation—specifically the prevalence of advertising, paywalls, and fragmented data sources in favor of direct, synthesized responses. Leaving out mainstream ones like IMDb, Letterboxd, Rotten Tomatoes and general purpose AI chat bots, other applications that target the idea of movie recommendation, description and ranking, exist already in a similar focused fashion like the one presented in this thesis, although they remain small projects than actual commercial apps. Some examples would be VivekaAryan/Movie$_r ecommendation_s ystem()$ both of them being very useful for finding the right movie, even though they may need to be configured in order to get them to work.

So why do we need a platform specific to a certain use case topic? Because of data accuracy! Despite the versatility of general-purpose Large Language Models (LLMs), they often lack the domain-specific grounding required for high-stakes accuracy. This thesis argues that 'Platform-Specific AI Implementation' is necessary to mitigate the risks of hallucination and generic outputs. By utilizing targeted Prompt Engineering and structured data grounding (e.g., specific unsorted data sets), we can ensure that the AI's creative synthesis remains factually correct.

2.1 Advantages and disadvantages of existing solutions

To evaluate the necessity of a specialized semantic analysis tool, it is essential to weigh the strengths and limitations of current available applications, most importantly market leaders. Existing solutions for film discovery generally split into two categories: highly structured relational databases and unconstrained generative models. While both offer significant utility, neither successfully bridges the gap between raw data accuracy and qualitative, synthesized insight without significant user intervention.

2.1.1 IMDb

Introduction and Short History

The Internet Movie Database (IMDb) is the world's most comprehensive and authoritative source for film, television, and celebrity content. It originated in 1990 when British computer programmer Col Needham posted a collection of Unix shell scripts to a Usenet discussion group, allowing users to search lists of actors and directors. By 1998, recognizing its immense potential as an information directory, Amazon.com acquired IMDb. Today, it stands as the industry standard for cinematographic metadata.

Data Handling and Searching

IMDb operates as a highly structured relational database. Its data is crowdsourced from industry professionals and verified volunteers, ensuring a massive, constantly updated repository of titles, release years, cast, crew, and technical specifications. Searching on IMDb is strictly deterministic and keyword-based. When a user searches for a title, the system queries its database and returns isolated tables of facts. User ratings are aggregated using a proprietary weighted average formula rather than a simple arithmetic mean, which aims to prevent rating manipulation and improve ratings accuracy.

Advantages

- IMDb's primary strength is its unmatched data integrity and size.
- It provides an exhaustive, factual ground truth that is highly trusted by both casual viewers and industry professionals.

Disadvantages

- The platform suffers from "Data Siloing"(data is contained in isolated repositories and is controlled by the IMDb organization) and high search friction.
- It does not natively synthesize information; to understand the "creative pedigree" of a film, a user must manually click through the director's page, cross-reference previous collaborations with the lead actors, and mentally calculate their historical success.
- It lacks qualitative synthesis.

2.1.2 General-Purpose Conversational AI: The Case of ChatGPT

Introduction and Short History

General-purpose conversational AI models, most notably ChatGPT (developed by OpenAI and launched in late 2022), represent a massive paradigm shift in information retrieval. Built on Large Language Model (LLM) architectures, these systems are trained on the vast data of the internet in order to generate human-like text responses based on user prompts.

Data Handling and Searching

Unlike IMDb, ChatGPT does not query a structured database of facts. Instead, it relies on its pre-trained neural network to generate probabilistic responses. When a user asks for movie recommendations or an analysis of a film, the AI synthesizes its training data to output natural language summaries, comparisons, and conversational advice.

Advantages

- ChatGPT drastically reduces search friction.
- It can instantly synthesize complex queries, identify thematic similarities ("vibes"), and provide a qualitative verdict without requiring the user to navigate multiple web pages.

Disadvantages

- Its greatest flaw in this context is the risk of hallucination.
- Because it generates text probabilistically rather than pulling from a rigid relational database, it can invent non-existent cast members, confuse release dates, or falsely claim that a director and actor have collaborated before.
- Unless connected to live data, its knowledge cutoff prevents it from accurately analyzing recent releases.

2.1.3 Specialized LLM Recommendation Systems: The Case of VivekaAryan Movie recommendation system

Introduction and Background

To bridge the gap between traditional databases and AI, developers frequently build specialized machine learning pipelines. A notable open-source example is the movie recommendation system developed by VivekaAryan on GitHub. This project represents a modern, open-source approach to utilizing AI for cinematic discovery.

Data Handling and Searching

This architecture abandons traditional keyword search in favor of vector-based semantic search. It uses the Sentence Transformer library to convert movie plots and metadata into mathematical vectors (embeddings). These embeddings are stored and queried using Weaviate (a vector database) and orchestrated via LangChain. Finally, it utilizes Phi-3 (a lightweight LLM) to generate natural language summaries of the retrieved movies.

Advantages

- By using semantic vector embeddings, the system is excellent at finding movies with similar plots and underlying concepts, even if they do not share exact keywords or genres.

- Using a local model like Phi-3 also provides cost-effective summarization.

Disadvantages

- While highly technical, this system focuses heavily on content/plot similarity rather than "creative pedigree."
- Relying on heavy vector databases introduces significant architectural overhead, and it does not explicitly weigh the historical success (ratings/track record) of the human crew behind the film.

2.2 Comparative Analysis and the Proposed Solution

The table below summarizes the operational differences between these existing methodologies and the solution proposed in this thesis:

Table 2.1: Comparative Analysis of Information Retrieval Methodologies

Feature	**IMDb (Traditional DB)**	**ChatGPT (General AI)**	**VivekaAryan (Vector/LLM)**	**Proposed Solution (Data-Grounded LLM)**
Data Architecture	Relational / Tabular	Neural Weights	Vector Database (Weaviate)	API Integration (Kaggle/IMDb + Gemini)
Search Paradigm	Keyword Matching	Probabilistic Generation	Semantic Embedding Similarity	Heuristic Prompt Engineering
Factual Accuracy	Very High	Variable (Hallucination risk)	High (within embedded dataset)	Very High (Anchored to IMDb API)
Qualitative Synthesis	None (Manual User Effort)	High (But ungrounded)	Moderate (Focuses on plot summaries)	High (Focuses on "Creative Pedigree")

Existing solutions force a compromise: users must choose between the strict, disconnected facts of IMDb, the conversational but potentially inaccurate nature of ChatGPT, or the plot-focused complexity of vector-based systems.

The proposed application resolves this by utilizing a Data-Grounded LLM approach. By first extracting verifiable, semi-structured metadata from the Kaggle API (acting as the ground truth), and explicitly feeding that verified data into the Gemini API using targeted prompt engineering, the system eliminates the hallucination risk of general AI. Furthermore, unlike the VivekaAryan model which focuses on plot embeddings, the proposed system's heuristic framework is specifically designed to analyze the collaborative history and track record of the cast and

crew. This ensures the user receives a highly accurate, qualitative "Verdict" and "Pedigree Analysis" with zero search friction.

Chapter 3

Some Latex Commands

In this chapter are presented some useful Latex commands that could be used in thesis written part.

3.1 Enumeration usage

Enumeration with numeric label:

1. Option 1
2. Option 2
3. Option 3

Enumeration without label:

- Option 1
- Option 2
- Option 3

3.2 If you need to add: definitions, theorems, ...

Definition 3.2.1. ...

Theorem 3.2.1. ...

Proof. ... □

Remark 3.2.1. ...

Example 3.2.1. ...

3.3 Bibliography References

The bibliography has to be referenced in thesis content using cite (e.g. [Knu98]).

3.4 Figure

Each figure has to have a caption that is a suggestive description of what the picture represents (e.g. Figure 3.1).

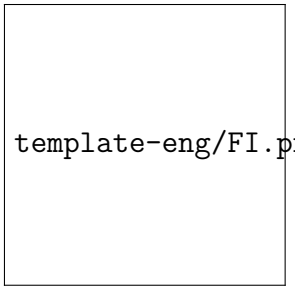

Figure 3.1: FMI logo scaled at 25% of text width

3.5 Algorithm pseudo-code

Pseudo-code is a formal way to describe an algorithm, is more clear than a textual description ore code (e.g. Algorithm 1).

Algorithm 1 An algorithm with caption

Require: $n \geq 0$
Ensure: $y = x^n$
1: $y \leftarrow 1$
2: $X \leftarrow x$
3: $N \leftarrow n$
4: **while** $N \neq 0$ **do**
5: **if** N is even **then**
6: $X \leftarrow X \times X$
7: $N \leftarrow \frac{N}{2}$ ▷ This is a comment
8: **else if** N is odd **then**
9: $y \leftarrow y \times X$
10: $N \leftarrow N - 1$
11: **end if**
12: **end while**

3.6 Source Code

If the thesis contains source code from the application, do not use print-screens, or verbatim use listing (e.g. listing 3.1).

Listing 3.1: An exemple of Python code

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
```

```
4       m = len(genl1)
5       n = len(genl2)
6       M = None #to become the incidence matrix
7       VT = np.zeros((n*m,1), int)  #dummy variable
8
9       #compute the bitwise xor matrix
10      M1 = bitxormatrix(genl1)
11      M2 = np.triu(bitxormatrix(genl2),1)
12
13      for i in range(m-1):
14          for j in range(i+1, m):
15              [r,c] = np.where(M2 == M1[i,j])
16              for k in range(len(r)):
17                  VT[(i)*n + r[k]] = 1;
18                  VT[(i)*n + c[k]] = 1;
19                  VT[(j)*n + r[k]] = 1;
20                  VT[(j)*n + c[k]] = 1;
21
22                  if M is None:
23                      M = np.copy(VT)
24                  else:
25                      M = np.concatenate((M, VT), 1)
26
27                  VT = np.zeros((n*m,1), int)
28
29      return M
```

3.7 Tablels

A simple example of a table (see Table 3.1).

Stopping criteria	**Alg.1**	**Alg.2**	**Alg.3**
MSQ	0.97	0.8	00.60
R2	0.77	0.78	0.54

Table 3.1: Algorithm comparison

For more details about how to create a table, use the following reference `https://www.overleaf.com/learn/latex/Tables`.

Bibliography

[Knu98] Donald E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, 3 edition, 1998.

Appendix A

Appendix - Generative AI Tools Usage

An overview how generative AI Tools are used in this thesis is presented in Table A.1.

AI Tool	Use case	Used in	Remarks
ChatGPT 5.0	Generate functionality list Generate initial state of the art bibliography list	Chapter 1, section 1.3 Chapter 2	Prompts in section A.1 Prompts in section A.1
Microsoft Copilot	Code generation	Chapter 4 pp.2-21	Details in section A.3
Deep L	Translation of text pages	Chapter 2	Prompts in section A.2
Grammarly	Correct grammar and style mistakes	Entire work	Free version
...			

Table A.1: Overview of generative AI Usage

A.1 ChatGPT Details

Specify for each usage the following:

1. Purpose of use
2. Method (which promts and for which parts of the thesis)
3. How the output has been examined/verified (has the generated output been examined for accuracy and reliability?)
4. How the output is implemented/used

A.2 Deep L Details

A.3 Microsoft Copilot