BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026
http://excel.fit.vutbr.cz

Intelligent Data Integration from Web Sources

Bc. Rudolf Jurišica*

Abstract

Due to the vast amount of inconsistent, unstructured, and duplicated data in web sources, it is essential to convert such data into a unified format. This paper addresses this challenge by processing car advertisements from the Czech web portals bazos.cz and hyperinzerce.cz. Using locally operated Large Language Models (LLMs), information is extracted from these listings. Searching within the processed advertisements is enabled in a web application through either a standard parametric form or a module utilizing Retrieval-Augmented Generation (RAG).
The system is designed to be configurable, allowing for expansion to other web sources. This creates a comprehensive system that aggregates offers from numerous sources into a unified form.

*xjuris02@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Web sources are primarily designed for human understanding. However, with the significant growth of data on the internet, it is becoming necessary for search engines, aggregators, and other automated tools to understand these pages as well. While some sources contain structured information, most do not.

This work deals with unstructured data from web sources, targeting Czech car classifieds portals – bazos.cz and hyperinzerce.cz. These sites were chosen because no existing platform currently transforms their data into a structured format, despite Bazoš being one of the largest portals with approximately 70,000 car listings. Furthermore, automobiles possess many distinct parameters, making them ideal for comparison based on extracted information.

The proposed solution processes listings from the aforementioned websites and converts them into a structured format using Large Language Models (LLMs), enabling efficient interaction with the listings (e.g., searching based on parameters). A module utilizing Retrieval-Augmented Generation (RAG) is integrated to allow for natural language search within the processed listings.

2. System overview

The developed system can be represented as a computational pipeline consisting of several steps: retrieval, information extraction, parameter enrichment, storage, and finally, user interaction – whether through a standard web application search or a chat interface for natural language queries. This entire process is divided into five main stages.

2.1 Data retrieval

The first step involves collecting information from the web sources bazos.cz and hyperinzerce.cz. Figures Fig. 1 and Fig. 2 show examples of listings on these platforms. Both contain unstructured information within the listing description (e.g., engine displacement, power, brand, car model ...).

The implementation was carried out in the Go programming language using the Colly library, as it offers ease of use and high performance when scraping large volumes of listings. Table 1 presents a comparison of the scraping speed for all listings from bazos.cz for Mazda and Audi vehicles. The Colly library achieved significantly higher speeds than the commonly used BeautifulSoup library and was therefore chosen for this work.

2.2 Information extraction

After retrieving the listings from the web sources, the required information must be extracted. This is achieved using Large Language Models (LLMs) running locally via the Ollama tool. Leveraging the high computational power of a reference machine equipped with an RTX

A5000 24GB GPU, the `qwen3:30b-instruct` model was utilized. This model demonstrated the best inference speed to extraction quality ratio (for more details, see 3).

The LLM is provided with a prompt that employs *Few-shot prompting* and *Persona* principles. The task involves extracting predefined parameters (e.g., engine displacement, power, fuel type, as shown in Fig. 3), while simultaneously performing data normalization and derivation (see Fig. 4). To ensure the accurate extraction of the brand and model, a two-phase approach is applied. First, an initial prompt is used to extract the brand (the prompt includes a list of possible brands from which the LLM must choose). Subsequently, a second prompt extracts the model (providing a list of valid models for the identified brand) alongside the remaining parameters. This strategy prevents the LLM from generating non-existent or unnormalized brands and models (e.g., outputting *Golf GTI* instead of *Golf*).

2.3 Data extending

To obtain as much information as possible about the processed data (the listed vehicles), it is highly beneficial to enrich the dataset with additional parameters (see Fig. 5). This enables potential buyers to filter and select vehicles based on specifications that are not explicitly mentioned in the original listing.

For this purpose, a vehicle catalog was created, comprising 132 brands, 1,652 models, and a total of 45,793 vehicle modifications produced between 1899 and 2026. During the processing of a listing, the identified car is matched to a corresponding catalog record. This retrieves supplementary vehicle specifications – such as curb weight, acceleration, average fuel consumption, production years, and more.

Furthermore, the vehicle catalog serves as a knowledge base for the AI assistant when generating responses, effectively acting as a preventative measure against hallucinations (for more details, see 2.4).

2.4 AI assistant

The AI assistant module offers convenient natural language search within the processed listings. Additionally, thanks to its conversational capabilities, it can serve as an advisor when choosing a car.

It is built on the *Retrieval-Augmented Generation* (RAG) principle, where the information sources are either the listings database, the vehicle catalog, or the general knowledge of the given LLM. The module is configurable, allowing to use OpenAI, DeepSeek, or locally running models via the Ollama tool (Fig. 6).

Depending on the user's query, the main agent decides which processing strategy to employ. There are 7 available strategies:

- *SQL* – simply generating an SQL query and retrieving data from the listings database
- *Semantic* – semantic search using vector similarity within the listings
- *Hybrid* – a combination of an SQL query and semantic search
- *Catalog* – searching exclusively within the vehicle catalog
- *Aggregate* – a statistical query to the database (e.g., count, minimum, average ...)
- *Clarify* – the agent lacks sufficient information and needs to ask the user for clarification
- *Conversational* – a standard response when the user's message is not related to automobiles

2.5 Web application

Interaction with the processed listings is facilitated by a web application. Searching can be performed either using specific parameters (a form-based search) or via the AI assistant (2.4). This AI interaction is handled through a chat interface (Fig. 7), which optionally displays listing previews alongside links to either the vehicle catalog or the listings database.

3. Experiments

Various combinations of prompt variants were tested on models of different sizes to evaluate their output quality. Four variants were selected (those identified as the most effective according to a 2024 study [1]). The results (Table 2) showed that there is no universally best prompt – the optimal one must always be determined individually for a specific LLM.

4. Conclusions

In this work, a comprehensive system was developed for the processing and presentation of unstructured data from web sources. It can be easily extended to include additional data sources. Data retrieval and information extraction can be executed periodically to keep the data up to date. Furthermore, experiments demonstrated that there is no universally optimal approach to prompt engineering; instead, prompts must be individually tailored to each specific model.

Acknowledgements

I would like to thank my supervisor, doc. Ing. Radek Burget, Ph.D., for his regular consultations, valuable advice, and for providing guidance on the direction of this thesis.

References

[1] Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4, 2024.