

LociGraph: Al Agent Framework for Browser-Based Knowledge Graph Construction



Nathan Cho | Advisor: Rose Sloan | Computer Science Program | Senior Project, Spring 2024

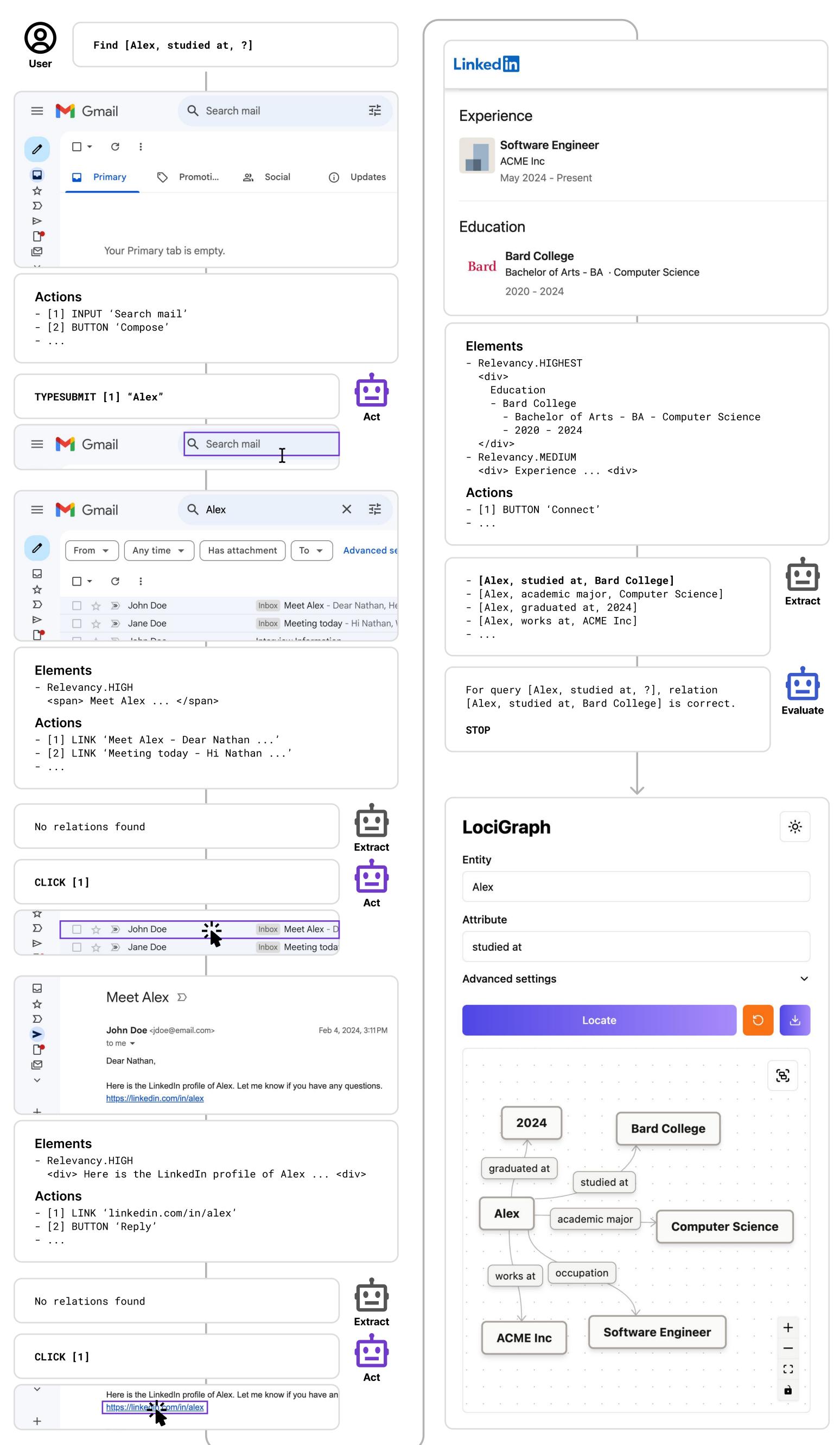
Introduction

Each person interacts with hundreds of webpages every day. Between hundreds of webpages, information is stored in hundreds of different unstructured formats (e.g., email, messages, posts, and social media profiles). This lack of structure and separation of data makes it very difficult to manage information in a centralized and organized way.

While information on the public web (e.g. Wikipedia) can be easily accessed using search engines, non-public webpages (e.g., email inbox, online community, social media) need to be manually visited to search.

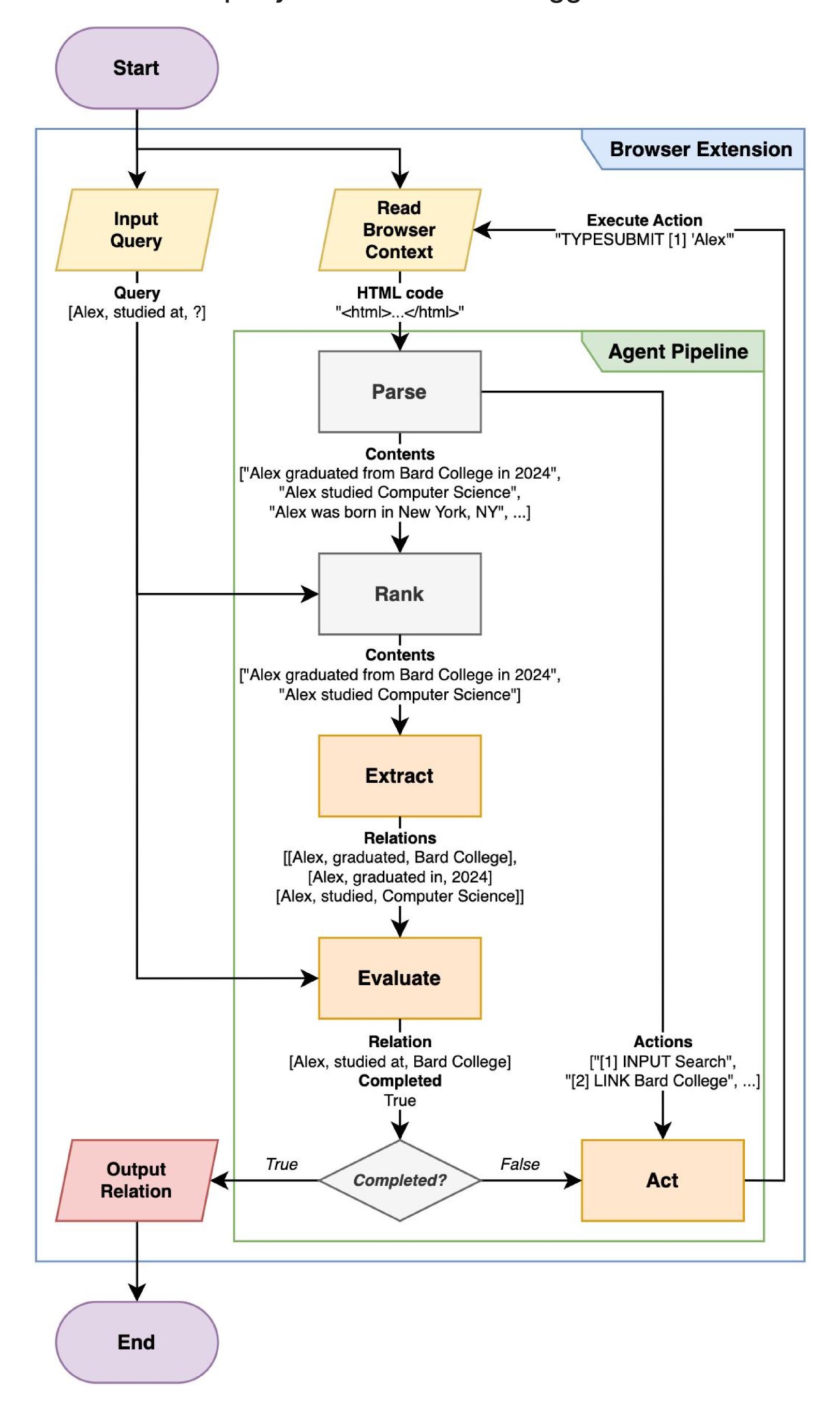
Motivated by such manual information-searching tasks, this work aims to create a system that can autonomously complete the task of **browser-based knowledge graph construction**, defined and shown in the example below.

- 1. System is given a query and access to a web browser.
- 2. System navigates to a webpage related to the query.
- 3. System extracts the relevant information into a knowledge graph.



Architecture

The framework consists of two parts: an *agent pipeline*, where agents analyze the webpage content and suggest the next action, and a *browser extension*, where the user can enter the query and execute the suggested action.



The agent pipeline consists of 5 subtasks: webpage parsing, element ranking, relation extraction, result evaluation, and action prediction. Compared to general-purpose web navigation agents, this framework adds extra steps of element ranking and result evaluation to reduce the token cost and improve the correctness of the result.

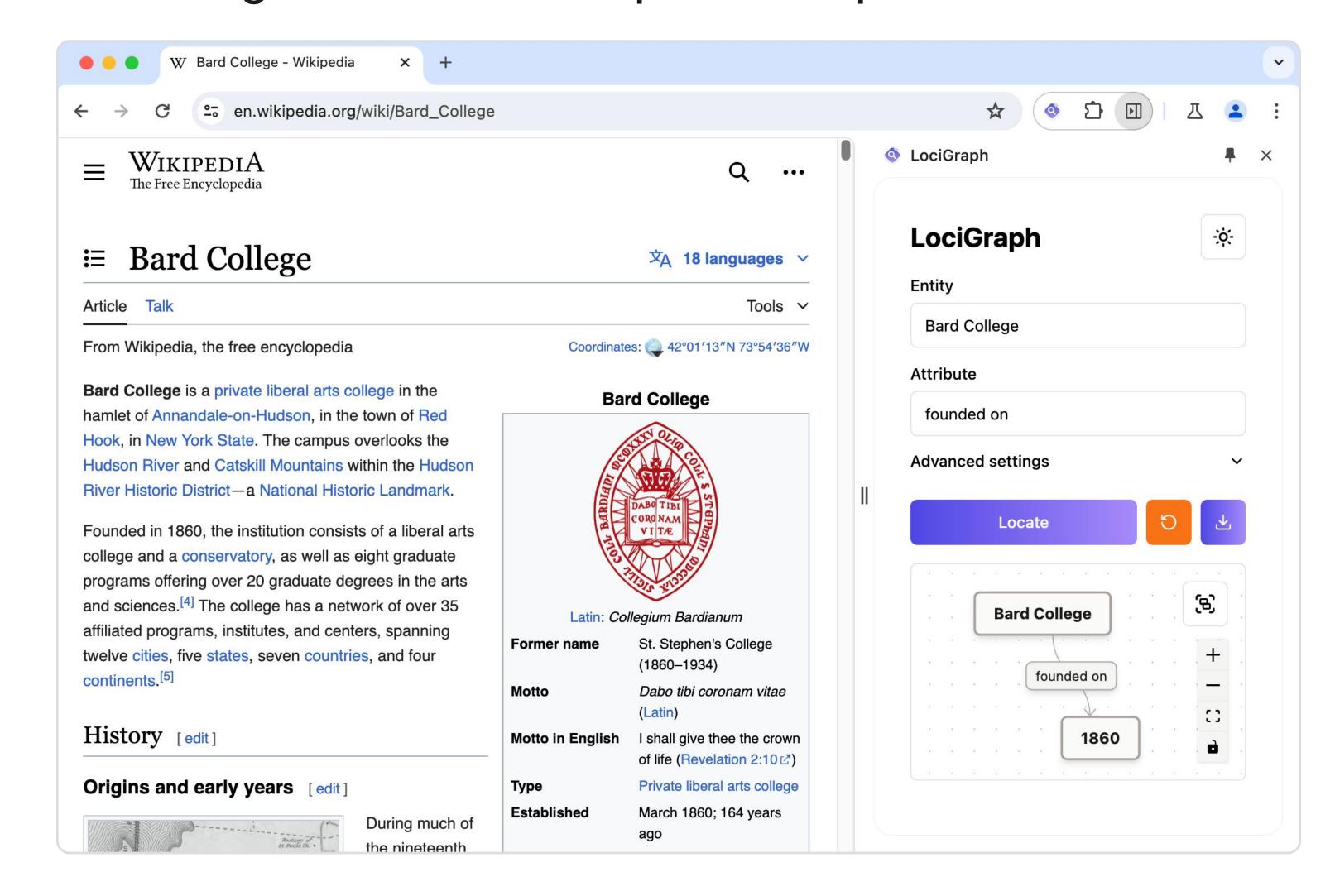
The *element ranking* task ranks static and interactive elements on the webpage by their relevancy to the query, determined by the text content, attributes, and location in the document. This ensures the pipeline to only process the relevant content, instead of the entire webpage.

The *result evaluation* task evaluates whether the extracted results are correct and satisfactory enough to answer the query, and determines whether the framework should continue navigating to a different webpage.

Implementation

The agent pipeline is implemented as an API server, receiving requests from the browser extension over HTTP connections and processing the pipeline in the server. This architecture of separating the pipeline from the browser extension minimizes the computational load on end users.

Compared to previous web agents built on browser testing tools, this work includes the implementation of a browser extension, exploring how the latest artificial intelligence technologies can be developed into a product.



Evaluation

The framework was tested on a small benchmark of 130 queries extracted from Wikipedia. The preliminary results show that large language model agents are capable of reasoning with websites represented in text, extracting information in a structured format, and navigating to relevant webpages. However, failure cases show the element ranking algorithm struggles to find indirectly relevant content using the keyword expansion.

Outcome	Count	Details
Success	127	Results contained a relation with a semantically correct target value.
Failure in rank	2	The rank process failed to find the relevant element due to keyword expansion not matching the element content.
Failure in extract	1	The extract agent hallucinated results due to lack of relevant contents.

Video demo is available on the homepage: https://ntcho.github.io/LociGraph



Future Work

- Using screenshots with large multimodal models
- Improving LLM agents with fine-tuning, multi-agent cooperation, automatic chain-of-thought prompting
- Improving element ranking with word similarity search and pre-trained language models for HTML code
- Integrating with knowledge graph databases