CTU Journal of Innovation and Sustainable Development
ISSN 2588-1418 | e-ISSN 2815-6412

DOI:10.22144/ctujoisd.2024.323

An automated data collection process for constructing graph data relying on LLMs

Ho Ngoc Ton, Nguyen Hoang Son, Nguyen Ngoc Minh Chau, and Pham-Nguyen Cuong*
University of Science, Vietnam National University Ho Chi Minh City, Viet Nam
**Corresponding author (pncuong@fit.hcmus.edu.vn)*

Article info.

Received 20 Jun 2024
Revised 1 Sep 2024
Accepted 2 Oct 2024

Keywords

Data collection process, graph data, large language model

ABSTRACT

This paper introduces a process that is designed to harvest data automatically from a variety of online sources. The core of this process lies in its data-handling techniques, which include drawing, cleaning, deduplicating, extracting, and categorizing of raw data to convert unstructured data into a structured format represented and imported in a graph database. The data extraction step utilizes Large Language Model (LLMs) for Named Entity Recognition (NER). A case study on deploying course data collection illustrates the enhancements brought about by this automation, showcasing improvements in the accuracy, completeness, and timeliness of updates in the course data. An evaluation carried out on the extraction and matching methods shows that the F1-score and precision rates are high. Overall, this study contributes to advancement of the field by providing a methodology for automating the collection and processing of online data sources, significantly improving the quality of data collection from online sources.

1. INTRODUCTION

In the rapidly evolving domain of online data sources, the integration of data from multiple origins has become standard practice, particularly with the advent of cloud computing and big data technologies. Traditional methods for supplying data to these systems predominantly rely on manual collection and processing techniques. These conventional approaches are not only labor intensive but also susceptible to inaccuracies and inefficiencies (Ranjith et al. 2022). Such limitations significantly impede the ability of applications to deliver timely, updated, and comprehensive guidance when aggregating and managing information.

As the repository of online data sources expands to encompass an increasingly diverse array of platforms, the demand for automated mechanisms to streamline and optimize these processes has become critical. Some existing approaches to addressing this issue involve the construction of solutions that rely on manual intervention. However, manual or semi-automated data management is not only time-consuming but also prone to human error, resulting in substantial inconsistencies and informational gaps within the datasets (Zhao, 2017). These gaps and inconsistencies undermine the integrity of the algorithms and systems reliant on this data for processing.

This study focuses on the critical need for an automated method capable of collecting, cleansing, and classifying course data from a variety of sources. It endeavors to improve systems and applications that consume data by introducing automation, harnessing the capabilities of NLP, and prioritizing system efficiency. NLP empowers the system to extract insightful information from unstructured text data, enhancing the accuracy and contextuality of the data (Wang et al., 2023). Integrating NLP capabilities into the system constitutes a pivotal step towards enabling it to comprehend and interpret textual information.

The main contribution of this paper is to propose a process framework that automatically collects data from online sources. This process consists of data retrieval, deduplication, extraction and storage. It employs rules to identify and remove duplicated entries and incorporates GPT-3.5, an LLM, to analyze collected data, extract and update the set of technological terms. A case study of online courses was implemented to illustrate how our process framework can be instantiated in a real-world environment.

The paper is organized as follows: Section 2 summarizes theoretical backgrounds. Section 3 discussed related works. Section 4 describes the motivation of our proposed approach. Section 5 focuses on our process framework for collecting data, which includes data collection, deduplication, analysis, and processing modules. Section 6 demonstrates a case study implementation for collecting online courses. Section 7 presents the evaluation of the approach and finally, Section 8 highlights main results and provides some perspectives.

2. THEORETICAL BACKGROUNDS

This section describes the theoretical backgrounds and technologies used in every step of the collection process. It also discusses and analyzes how methods are selected for implementation.

2.1. Web scraping techniques

Web scraping is a widely utilized method for obtaining data from the web due to its efficiency and efficacy. It involves the automated retrieval of data from websites to extract valuable information, which can then be integrated into a structured database (Zhao, 2017). This technique is particularly beneficial for managing dynamic domains such as job listings, course catalogs, user profiles and others, where new content is frequently added and updated.

Websites host vast amounts of data that can be utilized to collect information, addressing analytical and statistical requirements arising from substantial daily data generation. Accessing the Internet via a web browser is advantageous as it presents data visually, facilitating information perception and collection. However, the data displayed on a webpage is limited to what can be viewed in the browser. A web scraper, on the other hand, is a technology that enables the rapid gathering of data from thousands or even millions of sites, significantly outpacing manual methods (Xie, 2023). The process of collecting online data can be categorized into three primary stages (Nguyen et al., 2022): the retrieval stage includes accessing to websites to collect content using HTTP protocol; extraction stage involves extracting keywords, terms from the content; transformation stage converts keywords, terms into a structured format such as CSV, JSON, XML suitable for either presentation or storage, or incorporating it into a database for additional analysis or utilization.

Several tools are available, including Scrapy, Selenium, Requests, and Beautiful-Soup. Selenium is robust in scraping for web browsers, excelling in its capacity to process dynamic web content. The Requests library is noted for its user-friendly interface for initiating HTTP queries, while BeautifulSoup is particularly adept at parsing and navigating HTML and XML documents. Among these, Scrapy emerges as the most comprehensive solution, providing a framework for efficient web crawling, data extraction, and processing. It integrates HTTP requests with data extraction functionalities within a single package. Scrapy's advanced in managing complex websites, asynchronous tasks, and data pipelines. Its adaptability and scalability make it an optimal choice for most projects, especially those involving the processing of large data volumes.

Our project entails continuous scraping of substantial data volumes. Based on this comparative analysis, Scrapy is determined to be the most suitable tool due to its extensive capabilities and crucially.

2.2. Deduplication mechanism

Deduplication is a data processing technique aimed at eliminating duplicate instances of recurring data within a dataset (Kranz & Bigelow, 2019). It is critical for maintaining data quality and consistency, especially in systems that aggregate information from multiple sources or undergo frequent updates. This mechanism ensures that analytical and decision-making activities are based on accurate and unique data points, making deduplication essential for establishing a clean, organized, and reliable data repository for any data-centric organization. Some existing mechanisms are: exact matching (Ma et al., 2017), fuzzy matching (Chaudhuri, 2003; Ranjith, 2022), rule-based deduplication (Jiang et al., 2014) and Machine learning-driven deduplication (Akbar et al., 2023). Exact matching identifies and removes data entries that are identical. It is particularly effective for identical duplicates caused by system errors or multiple imports. Fuzzy matching recognizes records that are not exact duplicates but share similarities, accounting for inconsistencies, such as misspellings and abbreviations. Rule-based deduplication uses customized logical or domain-specific rules to identify duplicates. This approach can be tailored to specific data attributes and relationships, providing precise control over the deduplication process. Machine learning-driven deduplication trains models on labeled datasets to identify duplication patterns. This scalable method adapts to evolving data characteristics and is suitable for complex deduplication tasks.

In summary, exact matching ensures absolute precision but may be unsuitable for handling variations and inconsistencies in data. Fuzzy matching offers flexibility but requires careful parameter tuning and can be computationally intensive. Machine learning can adapt to complex patterns but typically needs large training datasets and may lack transparency in its decision-making processes.

Rule-based matching stands out for its clarity, customizability, and efficiency in defining criteria for eliminating duplicate items. This mechanism allows for control over the deduplication process, making it a reliable technique, particularly in scenarios where explicit rules and conditions can be defined for data comparison and deduplication. This is especially advantageous when handling large datasets, ensuring rapid and accurate duplicate identification while maintaining data quality and integrity. Therefore, a rule-based deduplication method is the most suitable solution that meets our requirements.

2.3. Data extraction methods

To extract meaningful information from unstructured data, it is essential to identify and extract names and other significant entities. This leads to the NER problem, which focuses on detecting and categorizing named entities in text data. Solving the NER problem enhances data understanding and analysis, enabling valuable insights and the development of advanced natural language processing applications. Several methods are used to extract entities from text such as dictionary-based (Saha, 2020), rule-based (Eftimov et al., 2017), machine learning (Yuefan & Xiaolong, 2023), and LLMs (Huang et al., 2024; Villena et al., 2024). The dictionary-based method relies on pre-existing lists of entities defined in a dictionary, offering a straight-forward approach to identifying specific phrases. However, it may struggle with variations and newly introduced entities not included in the dictionary. The rule-based method provides transparency and customizability by allowing explicit definitions of rules or patterns for recognizing entities like names, organizations and locations, but it may have limitations in coverage and adaptability due to its reliance on predefined rules. The machine learning-driven method uses algorithms and models trained on annotated corpora to detect and categorize named entities. Techniques such as Conditional Random Fields (CRF), Hidden Markov Models (HMM), and advanced deep learning methods like Recurrent Neural Networks (RNNs) and Transformer-based models are commonly employed. It adapts to complex patterns and is more robust than dictionary-based and rule-based methods, by exceling in capturing intricate patterns and generalizing across various linguistic contexts, but it requires meticulously curated annotated training data and careful feature selection for optimal performance.

The recent use of LLMs has revolutionized NER tasks. Models such as BERT (Bi-directional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have demonstrated strong abilities in understanding and processing natural language, including recognizing and categorizing named entities in unstructured text. These models grasp complex language patterns, contextual dependencies, and semantic links, crucial for accurate NER. LLMs possess contextual comprehension and bidirectional language processing capabilities, enabling them to deduce the meaning and importance of named entities by analyzing the surrounding context. They offer flexibility in custom domains through fine-tuning and can process complex linguistic patterns, resulting in enhanced accuracy in NER. Although fine-tuning LLMs requires significant computational resources and annotated data, their versatility and performance make them a revolutionary solution for precise and contextually aware entity recognition. Due to these advantages of LLMs, we utilize GPT-3.5, an LLM, for data extraction which is described in the following sections.

3. RELATED WORK

Most existing approaches employ a semi-automatic mechanism and rule-based methods for data collection and extraction (Thi, 2020; Nguyen, 2022; Tin, 2023; Hien, 2024;) as summarized in Table 1. Thi et al. (2020) describe their work on building the JobPosting dataset and ontology models using a rule-based extraction for collecting and analyzing required job knowledge and experience in the IT field. Ontologies provide a structured framework to represent a knowledge base for organizing and classifying information, which enables clear characterization of concepts and their connections. Hien et al. (2024) employed a similar process for a course recommender system, extracting relevant information from a dataset consisting of courses rather than job postings. Nguyen et al. (2022) proposed a three-phase data collection process using deep learning technology to extract competence entities from online courses. They used graph data to represent the collected courses and store them in a graph database. In a different approach, Tin (2023) proposed a semi-automatic process for knowledge construction in smart systems, which are capable of recognizing context and providing appropriate services or solutions based on this context. Ontologies play a crucial role in organizing context data within smart service systems.

Another promising approach is the utilization of LLMs for NER tasks, as demonstrated in GPT-NER (Wang et al., 2023). In this work, the authors introduced a 3-step process to build prompts for integrating LLM to solve NER problems including: a) task description: tells LLM what its role is; b) Few-shot demonstration: controls the output format, provides evidence and references to make accurate predictions, and defines a series of result instances; c) input sentence: feeds the current input sentence into the LLM and expects the LLM to generate the output sequence according to the defined format.

Table 1. Summary of the recent approaches of data collection

Approach	Deduplication	Extraction	Automated process	Adaptable to bew data
Thi et al. (Thi et al., 2020)	No	Rules-based	No	No
Hien (2024)	No	Rules-based	No	No
Nguyen et al. (2022)	No	Deep learning	N/A	Yes
Tin (2023)	No	Machine learning	No	No
GPT-NER (Wang et al., 2023)	N/A	LLMs	N/A	Yes
Our approach	Yes	LLMs	Yes	Yes

4. MOTIVATION

The motivation behind this automation lies in the recognition that traditional methods of data analysis struggle to keep pace with the scale and complexity of online resources. Manual or semi-automated processing is slow, error-prone, and limited in scope. By leveraging recent technologies such as artificial intelligence, automation can quickly analyze large datasets, extract valuable insights without human intervention, and provide a knowledge base that is more accurate and current.

As presented in Table 1, existing studies have shown positive outcomes, highlighting the efficiency of their methods in addressing challenges in the field. However, most of these studies rely on semi-automated processes, requiring manual intervention by domain experts at various stages, such as data labeling and rule definition. While the results are commendable, the need for manual handling during critical stages emphasizes the necessity for further advancements toward fully automated solutions. Future research should focus on simplifying and automating these manual processes to achieve more efficient and scalable solutions.

In this context, our research aims to incorporate LLMs into our extraction stage, leveraging their advanced capabilities to improve the extraction of key insights from unstructured data. The integration of LLMs is anticipated to enhance the precision and effectiveness of our extraction techniques.

5. DATA COLLECTION PROCESS DESCRIPTION

Data collection systems with crawling and extraction components are known as data acquisition systems. In our approach, they involve an automated collection process which is outlined in Figure 1, comprising three main modules which are data collection, data deduplication, and data analysis and processing. Each module is subdivided into several components. These components will be elaborated upon in the subsequent sections.

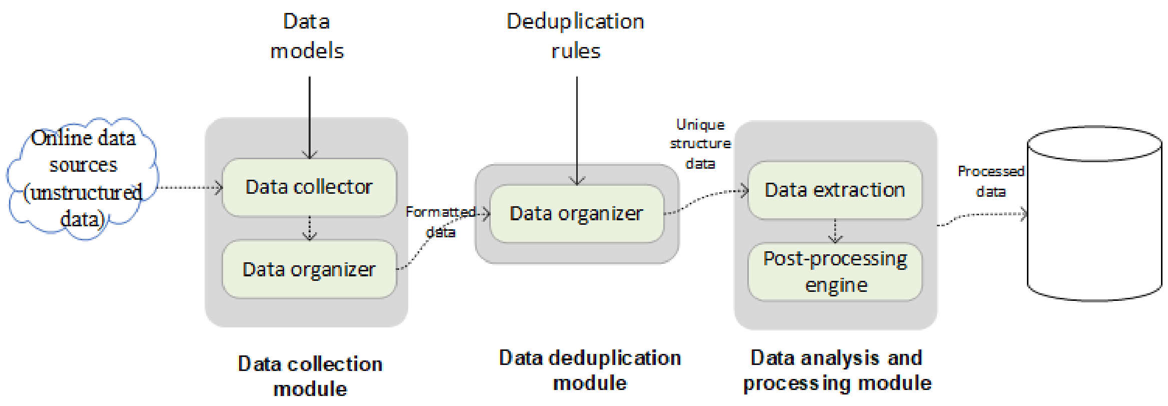


Figure 1. Overview of automated process for data collection

5.1. Data collection module

The purpose of data retrieval is to gather, pre-process, and transfer data from documents or other data sources into a structured format. These collected data files will be loaded in appropriate form suitable for processing. The module consists of two main components as follows:

– Data collector: This component gathers unstructured data from various online sources, enabling the collection of a wide array of disorganized data, including text-based content and multimedia. It consolidates unstructured data for the subsequent processing and analysis module.

– Data organizer: This component organizes unstructured data according to predefined data models set by domains. It classifies and structures the gathered data to meet specific domain criteria, arranging it into these models for seamless integration into subsequent phases of analysis and application development.

As mentioned previously, the Scrapy framework (Xie, 2023) is utilized for both data collector anf data organizer due to its advanced features.

5.2. Deduplication module

The data deduplication module is designed to identify and eliminate duplicate or redundant entries within the collected dataset. This module compares and analyzes the data to detect and remove any redundant instances. By eliminating duplicate entries, it ensures that subsequent analysis and processing of the data are based on accurate and non-repetitive information.

The module oversees a deduplication engine implemented using either rule-based or machine learning methods, as mentioned in Section 2.2. It is tailored to address the specific requirements and nuances of the dataset, enhancing its ability to accurately identify and eliminate duplicate records.

5.3. Data analysis and processing module

The data analysis and processing module serves a vital role in converting unprocessed data into knowledge represented by graph data. It comprises two components:

— The information extraction component is designed to extract useful information and knowledge from structured data, especially attributes such as description or content. It incorporates algorithms to detect and extract data based on a pre-defined knowledge model.

— The post-processing component refines and improves the extracted data to complement the information extraction component. It is responsible for the thorough refinement and optimization of the collected data, ensuring that they are well-prepared for inserting into knowledge bases.

As mentioned in the section 3.3, GPT-3.5, an LLM for NER task is used due to its advantage in extracting data by considering the context and the semantic, thereby improving accuracy.

6. IMPLEMENTATION: A CASE STUDY OF ONLINE COURSES

This section demonstrates our approach for data collection process, as applied to the context of online courses. Initially, we describe the system architecture and the course data model, which serves as the data structure framework for the knowledge base. Then, we illustrate the implementation of the collection process.

6.1. System architecture

In this section, we provide an overview of the system design and architecture before describing the implementation of each component. We also present our deployment strategy, which encompasses both the frontend and the backend. The frontend is developed using ReactJS, while the backend is divided into five components, each running in a separate container managed by Docker Compose:

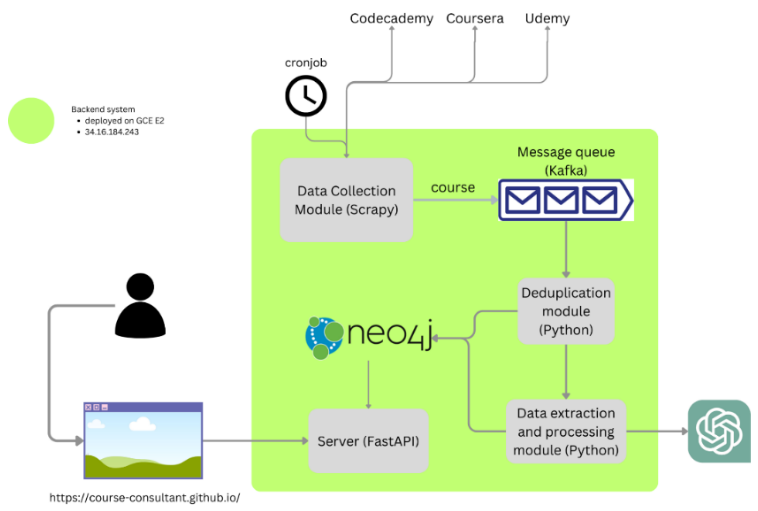

Codecademy
Coursera
Udemy
cronjob
Backend system
• deployed on GCE E2
• 34.16.184.243
Data Collection Module (Scrapy)
course
Message queue (Kafka)
Deduplication module (Python)
neo4j
Server (FastAPI)
Data extraction and processing module (Python)
https://course-consultant.github.io/

Figure 2. Overall system architecture

– Data collection module: this module is triggered by a weekly cronjob, continuously scrapes data from online learning websites, ensuring the knowledge base remains current.

– Message queue management (Kafka): this module acts as a message queue and intermediary communication bus between system components. Kafka supports asynchronous communication, enabling the processing of long-running jobs that cannot be handled via HTTP calls.

– Deduplication, extraction and processing module: Although introduced as separate modules based on their semantics and functionality, they are implemented within a single container due to their consecutive nature.

– Neo4j: Serves as a graph database platform to store the knowledge base.

– FastAPI: Utilized to set up a server and handle user requests.

6.2. Course model

The course model is reused and extended from the work of Nguyen et al. (2022) and Cuong et al. (2022). The model contains entities for learning courses and their relationships which are depicted in Figure 3. The entities are described as follows:

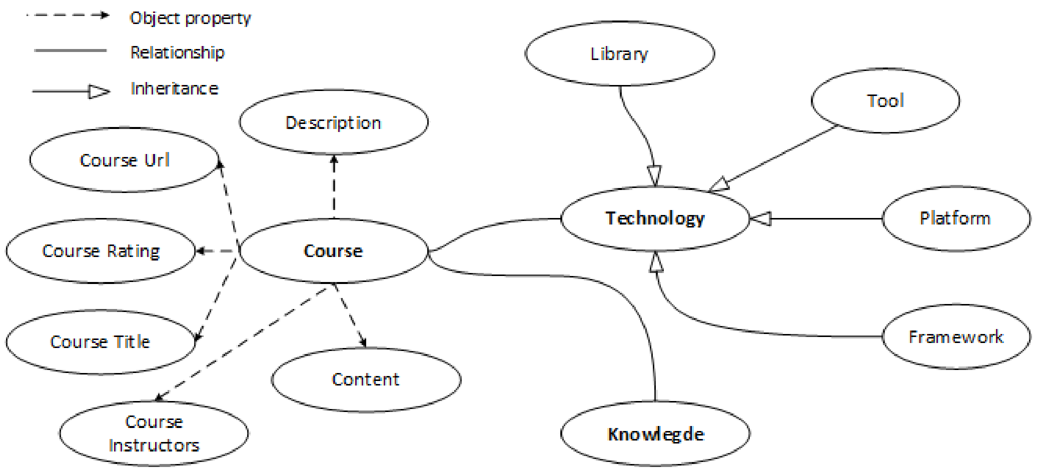


Figure 3. High level of course model

– Course: represents online courses collected from educational websites.

– Knowledge: refers to the fundamental concepts, theories, principles, or facts that students must be familiar with either before commencing the course or will acquire upon finishing it.

– Tool: refers to the skill in using particular tools or software related to the IT field. These requirements ensure that learners possess a specific degree of competency in utilizing these tools.

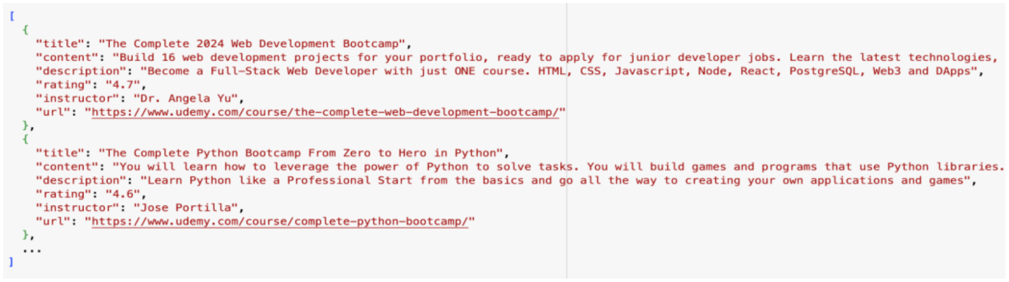
– Platform: involves the capability to operate and leverage various platforms effectively, such as operating systems, content management systems, or cloud platforms such as Azure or AWS.

– Framework and Library: refers to the requirements of the comprehension and utilization of different frameworks or libraries in programming.

This course model serves as a structured knowledge that allows standardization and labeling of data collected from online course websites.

6.3. Data collection implementation

The implementation of the courses collection is illustrated by instantiating the process described in Section 5. For the data collection module, we combine Crontab and Scrapy for scraping new data. The course scraper, which includes Udemy, Codecademy, and Coursera, is written in a single Python script using Scrapy. This script is configured to run weekly by scheduling it within a cron table, which maintains all system cron jobs. Scrapy is also employed for data extraction using both CSS and XML formats, offering a versatile mechanism for gathering and structuring information. This capability ensures the efficient retrieval of organized data from online sources. Additionally, Scrapy's functionality to export data in multiple formats enhances its simplicity and effectiveness for integration with the data organizer component.

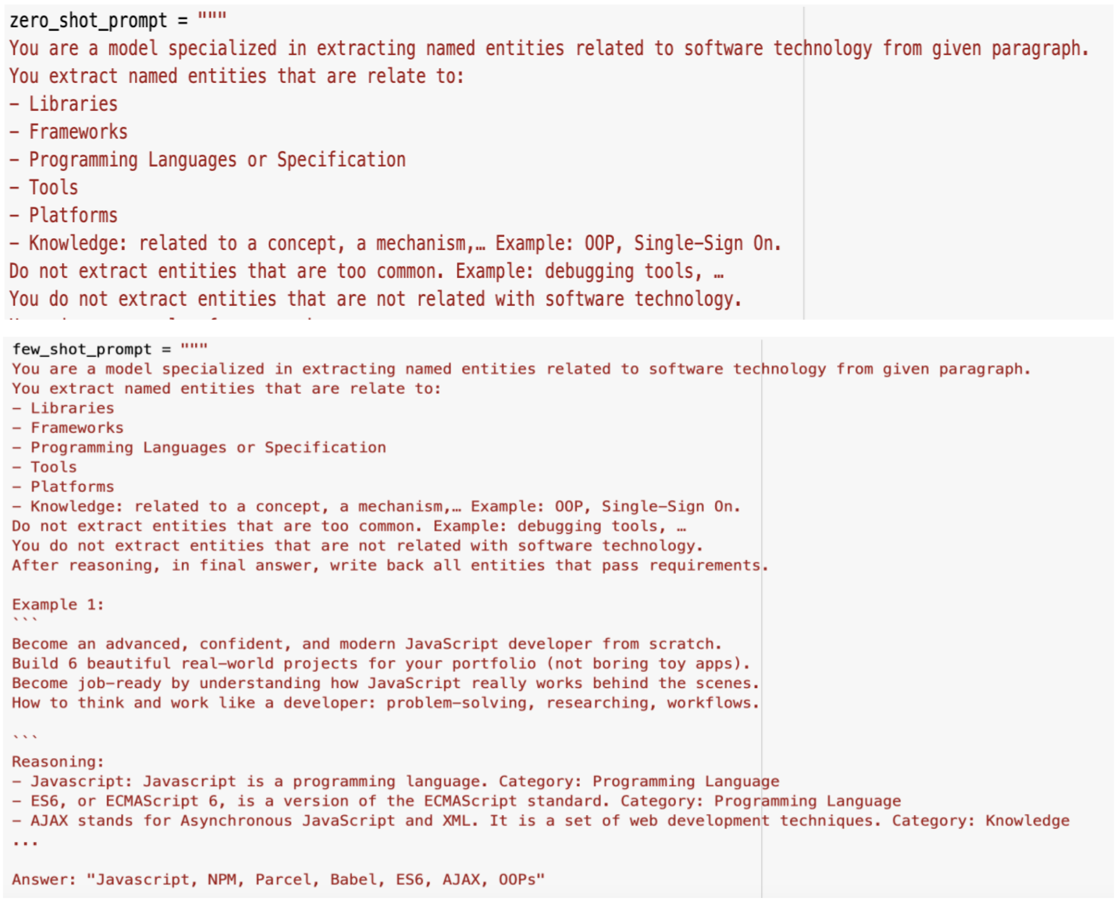


```
[
  {
    "title": "The Complete 2024 Web Development Bootcamp",
    "content": "Build 16 web development projects for your portfolio, ready to apply for junior developer jobs. Learn the latest technologies,
    "description": "Become a Full-Stack Web Developer with just ONE course. HTML, CSS, Javascript, Node, React, PostgreSQL, Web3 and DApps",
    "rating": "4.7",
    "instructor": "Dr. Angela Yu",
    "url": "https://www.udemy.com/course/the-complete-web-development-bootcamp/"
  },
  {
    "title": "The Complete Python Bootcamp From Zero to Hero in Python",
    "content": "You will learn how to leverage the power of Python to solve tasks. You will build games and programs that use Python libraries.
    "description": "Learn Python like a Professional Start from the basics and go all the way to creating your own applications and games",
    "rating": "4.6",
    "instructor": "Jose Portilla",
    "url": "https://www.udemy.com/course/complete-python-bootcamp/"
  },
  ...
]
```

Figure 4. Illustration of courses scraped and organized by the Scrapy spider

For the deduplication module, a rule-based approach is employed as predetermined criteria to identify and eliminate duplicate entries, thereby reducing storage costs and streamlining data processing. Each rule is formatted in an “if-then-else” structure and users can add new rules. With course data, rules are defined based on three attributes: CourseTitle, CourseURL, and CourseInstructor. Any new course entries with these three attributes matching an existing course in the dataset will be considered duplicates and removed.

Data extraction component is related to the skills extraction from learning outcomes which are entailed and parsed from course descriptions and content. For this step, we employ ChatGPT-3.5-turbo-1106 with two different prompts: Zero-shot and Few-shot as illustrated in Figure 5. Responses generated by ChatGPT are free text with no structure. To address this issue, we implemented OpenAI's function calling feature that enables GPT-3.5 to generate structured JSON data.



```
zero_shot_prompt = """
You are a model specialized in extracting named entities related to software technology from given paragraph.
You extract named entities that are relate to:
- Libraries
- Frameworks
- Programming Languages or Specification
- Tools
- Platforms
- Knowledge: related to a concept, a mechanism,… Example: OOP, Single-Sign On.
Do not extract entities that are too common. Example: debugging tools, …
You do not extract entities that are not related with software technology.
```

```
few_shot_prompt = """
You are a model specialized in extracting named entities related to software technology from given paragraph.
You extract named entities that are relate to:
- Libraries
- Frameworks
- Programming Languages or Specification
- Tools
- Platforms
- Knowledge: related to a concept, a mechanism,… Example: OOP, Single-Sign On.
Do not extract entities that are too common. Example: debugging tools, …
You do not extract entities that are not related with software technology.
After reasoning, in final answer, write back all entities that pass requirements.

Example 1:
```
Become an advanced, confident, and modern JavaScript developer from scratch.
Build 6 beautiful real-world projects for your portfolio (not boring toy apps).
Become job-ready by understanding how JavaScript really works behind the scenes.
How to think and work like a developer: problem-solving, researching, workflows.

```
Reasoning:
- Javascript: Javascript is a programming language. Category: Programming Language
- ES6, or ECMAScript 6, is a version of the ECMAScript standard. Category: Programming Language
- AJAX stands for Asynchronous JavaScript and XML. It is a set of web development techniques. Category: Knowledge
...

Answer: "Javascript, NPM, Parcel, Babel, ES6, AJAX, OOPs"
```

Figure 5. Simplified Zero-shot and Few-shot prompts

The post processing component is to resolve the data discrepancies and data inflation found from the data extraction component by using technological terms set and mapping the findings to this terms set. This approach aims to streamline and unify the technology terms, addressing inconsistencies and mitigating data inflation. To achieve this, fuzzy matching is employed using the Levenshtein distance algorithm (Yujian & Bo, 2007), which calculates the similarity between two strings by measuring the minimum number of single-character operations needed to transform one string into the other. This method facilitates the identification of keywords closest to our technology terms, allowing for merging or replacement of terms to establish standardized terms that align with the course model. In Table 2, the Levenshtein distance algorithm is applied with a threshold of 2, which optimally considers similar terms. In this instance, the terms “Python libraries” and “collection modules” are interpreted as wrongly retrieved and subsequently removed from the process. The term “GUIs” is determined to be closest to the standard term “REST APIs”, and is therefore replaced accordingly.

Table 2. Example of canonicalization steps using the Levenshtein algorithm

ChatGPT answer	Levenshtein distance with threshold = 2
Python, Python libraries, Python 2, Python 3, collections module, Object Oriented Programming, Jupyter Notebook, GUIs	Python, ~~Python libraries~~, Python, Python, ~~collections module~~, Object-Oriented Programming, Jupyter, REST APIs

Due to the advantages of graph databases in handling relationships through direct traversal, their schema flexibility, which makes them adaptable to changes in data models, the simplification of complex queries, and their scalability (Robinson et al., 2015), we have selected a graph database for our data storage approach. Furthermore, Neo4j is an open-source NoSQL database that specializes in graph data, offering ACID-compliant transaction support for applications. It is one of the most popular open-source graph databases available and is compatible with the cypher query language, which is known for its straightforward nature when querying graph-structured data (Singh, 2022).

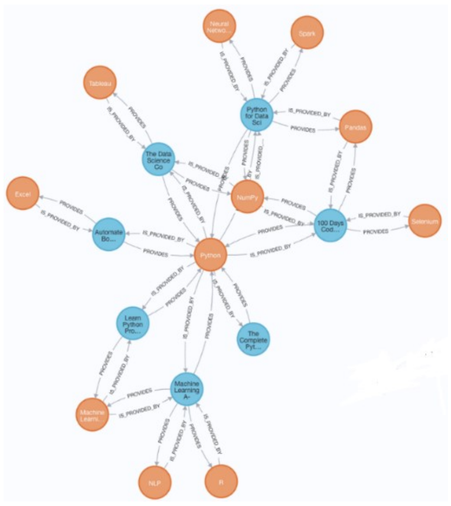


Figure 6. Illustration of segment of graph data representing courses

The collected data is represented as graph data and stored in the Neo4J graph database framework where each node signifies an entity such as a course or skill, while each edge signifies an attribute or relationship between nodes. Figure 6. demonstrates a segment of the graph database depicting courses.

7. EVALUATION

Our evaluation primarily focuses on the data analysis and processing module, which includes the extraction task using LLM and the mapping task to demonstrate their effectiveness in performing NER.

7.1. LLM extraction process

This section evaluates the capability of LLM in accurately extracting technology terms from a set of educational course descriptions, using precision, recall, and F1 scores as metrics for assessment. The test was executed on Google Colabs, connected to the T4 GPU provided by Google. The LLM model used is model gpt-3.5-turbo-1106. The dataset consists of 100 courses collected from the Udemy website. The technological labels manually extracted were 620. The matrices used for evaluation are shown in Table 3.

Table 3. Measurement of ChatGPT extraction

Features	True positive	False positive	False negative	Precision	Recall	F1-score
Zero-shot	494	112	126	80.2%	79.7%	79.9%
Few-shot	436	146	184	74.9%	70.3%	72.5%

Our observations indicate that the zero-shot prompting technique outperforms in both the accuracy of extraction and the number of correctly identified technology terms, with fewer omissions. This finding contradicts the prevailing assumption that more sophisticated prompting techniques, such as few-shot prompting, generally yield superior results in tasks requiring logical reasoning. Therefore, the zero-shot prompt is configured to use in our system.

7.2. Keywords mapping

This section presents the evaluation of the accuracy of the canonicalization process used to standardize technology terms extracted from course content. This involves testing how effectively the process translates varied technology keywords into a uniform set of standardized terms, using the precision metric to assess performance.

The testing dataset is generated using ChatGPT without any specific prompting providing 200 random technology keywords of any type. Each keyword was manually labeled with the correct term based on our standardized terms. Keywords that did not correspond to any standardized terms were labeled as "None”. Two evaluations are performed: the precision of mapping words to standardized terms and the precision of removing incorrectly retrieved words. Figure 7(a). shows that the highest precision is achieved when the threshold k =2, with k ranging from 0 to 10. Figure 7(b). presents the precision of removing incorrectly retrieved words, with the top three highest values achieved when the threshold k = 0, 1, and 2. The combination of these two evaluations indicates that the threshold k =2 is determined to be the optimal choice for this case. Table 4. presents the precision values when the threshold k = 2.

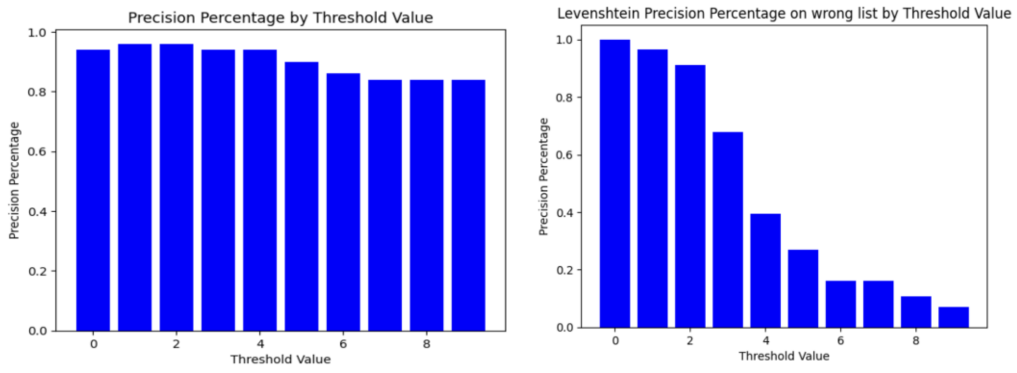


Figure 7. (a) Precision of Levenshtein distance; (b) Precision on incorrectly retrieved words.

Table 4. Summary of measurement using Levenshtein distance

Measurements	Precision
Correctly mapping words to standardized terms	92%
Remove incorrectly retrieved words	91%

8. CONCLUSION AND FUTURE WORK

In this paper, we explore the landscape of designing and implementing a process of collecting unstructured data from online sources which is based on LLMs for data extraction. We leverage advanced data processing techniques, featuring NER to analyze data. Our approach employs a suite of technologies such as Scrapy for efficient web scraping, Neo4j for robust graph database management, and FastAPI for crafting a responsive and user-friendly interface. A significant portion of our research was dedicated to experimenting with LLMs models for NER including terms extraction and keyword matching. A case study was conducted on an online course data set to illustrate the viability of the proposed process, which is scheduled to be carried out on a weekly basis to collect the necessary data. The findings indicated that the capacity of LLM to comprehend contextual nuances and extract pertinent entities/terms from unstructured text markedly enhanced accuracy, reaching a rate of over 79.9%.

In the future, we will continue to expand to various domains to assess the applicability across different sources of data. To achieve this, it is required to configure a data model, standard terms set and deduplication rules of a domain. Additionally, addressing the critical importance of processing performance will be a focus in our next steps. We are considering the use of additional big data platforms and technologies to improve performance.

ACKNOWLEDGMENT

This research is funded by the University of Science, VNU-HCM under grant number CNTT 2023-02.

REFERENCES

Akbar, M., Ahmad, I., Mirza, Ali, M., & Barmavatu, P. (2023). Enhanced authentication for deduplication of big data on cloud storage system using machine learning approach. *Cluster Comput*, *27*(3), 3683–3702. https://doi.org/10.1007/s10586-023-04171-y

Chaudhuri, S., Ganjam, K., Ganti, V., Motwani, R. (2003). Robust and efficient fuzzy match for online data cleaning. *In Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. Association for Computing Machinery, New York, NY, USA, 313–324. https://doi.org/10.1145/872757.872796

Cuong, N. D., Dung, D. N. H., Pham-Nguyen, C., Le Dinh, T., & Nam, L. N. H. (2022). Itcareerbot: A personalized career counselling chatbot. In *Asian Conference on Intelligent Information and Database Systems* (pp. 423-436). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-8234-7_33

Eftimov, T., Koroušić Seljak, B., Korošec, P. (2017). *A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations*. PLoS One, *12*(6), e0179488. https://doi.org/10.1371/journal.pone.0179488

Hien, P.T. X., Nam, L.N.H., & Pham-Nguyen, C. (2024). Framework for a knowledge-based course recommender system focused on IT career needs. *16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (KEOD 2024).

Huang, Y., Tang, K., & Chen, M. (2024). *Distilling Large Language Models into Tiny Models for Named Entity Recognition*. arXiv:2402.09282v3 [cs.CL].

Jiang, Y., Lin, C., Meng, W., Yu, C., Cohen, A. M., & Smalheiser, N. R. (2014). Rule-based deduplication of article records from bibliographic databases. *Database: The Journal of Biological Databases and Curation*, *2014*.

Kranz, G., & Bigelow, S.J. (2019). *Data deduplication*. https://www.techtarget.com/searchstorage/definition/data-deduplication.

Ma, J., Stones, R.J, Ma, Y., Wang, J., Ren, J., Wang, G., & Liu, X. (2017). Lazy Exact Deduplication. *ACM Trans on Storage (TOS). 13*(2), 1-26. https://doi.org/10.1145/3078837

Nguyen, T. M. T., Vu, N., & Ly, B. (2022). An approach to constructing a graph data repository for course recommendation based on IT career goals in the context of big data. *2022 IEEE International Conference on Big Data,* December 17-20, Osaka, Japan (pp. 301-308). 10.1109/BigData55660.2022.10020436

Ranjith, V., Dhananjaya, M.K., Sahukar, P.Y., Akshara, M., & Biswas, P.S. (2022). A Review of Deduplicate and Significance of Using Fuzzy Logic. In: Fong, S., Dey, N., Joshi, A. (Eds.) *ICT Analysis and Applications. Lecture Notes in Networks and Systems, 314*. Springer, Singapore. https://doi.org/10.1007/978-981-16-5655-2_27

Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph Databases* (2nd Ed.). O'Reilly Media, Inc.

Saha, S. (2020). *Biomedical Named entity recognition - Pros and cons of rule-based and deep learning methods*. https://www.cineca-project.eu/blog-all/biomedical-named-entity-recognition-pros-and-cons-of-rule-based-and-deep-learning-methods

Singh, A. (2022). *Graph Database Modeling With Neo4j* (2nd ed.). Independently published.

Thi, P.Q., Diep, H. T., Thao, N.D., Pham-Nguyen, C., Le Dinh, T., & Nam, L.N.H. (2020). Towards An Ontology-Based Knowledge Base for Job Postings. In: *7th NAFOSTED Conference on Information and Computer Science (NICS).* 267-272. VNUHCM-University of Science, Vietnam. November 26-27. https://doi.org/10.1109/NICS51282.2020.9335876

Tin, L.V. (2023). *Towards a context-aware ontology-based approach for knowledge discovery in intelligent systems.* University of Science, Ho Chi Minh city, Viet Nam (Thesis report).

Villena, F., Miranda, L., & Aracena, C. (2024). *llmNER: (Zero|Few)-Shot Named Entity Recognition, Exploiting the Power of Large Language Models*. arXiv:2406.04528 [cs.CL].

Wang, S., Sun X., Li X., Ouyang R., Wu F., Zhang T., Li J., & Wang G. (2023). GPT-NER: *Named Entity Recognition via Large Language Models*. arXiv:2304.10428[cs.CL]. https://arxiv.org/abs/2304.10428

Zhao, B. (2017). Web Scraping. In: Schintler, L., McNeely, C. (eds*) Encyclopedia of Big Data*. Springer, Cham. https://doi.org/10.1007/978-3-319-32001-4_483-1

Xie, Z. (2023). The benefit and risks for scraping based on Python. *Highlights in Science, Engineering and Technology*, *49*, 232-236. https://doi.org/10.54097/hset.v49i.8511

Yuefan, F., & Xiaolong, X. (2023). GFMRC: A machine reading comprehension model for named entity recognition. *Pattern Recognition Letters*, *172*, 97-105. https://doi.org/10.1016/j.patrec.2023.06.011

Yujian, L. & Bo, L. (2007). A Normalized Levenshtein Distance Metric. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1091-1095. 10.1109/TPAMI