


## Article

# LLM-Powered Natural Language Text Processing for Ontology Enrichment

Assel Mukanova <sup>1</sup>, Marek Milosz <sup>2,\*</sup>, Assem Dauletkaliyeva <sup>1</sup>, Aizhan Nazyrova <sup>1,3</sup>, Gaziza Yelibayeva <sup>3</sup>, Dmitrii Kuzin <sup>1</sup> and Lazzat Kussepova <sup>1</sup>

- <sup>1</sup> Higher School of Information Technology and Engineering, Astana International University, 8 Kabanbay Batyr Av., Astana 010000, Kazakhstan; assel.mukanova@aiu.edu.kz (A.M.); aizhan\_nazyrova@aiu.edu.kz (A.N.); dmitriy\_kuzin@aiu.edu.kz (D.K.); kusepova.liazat@aiu.edu.kz (L.K.)
- <sup>2</sup> Department of Computer Science, Lublin University of Technology, 36B Nadbystrzycka Str., 20-618 Lublin, Poland
- <sup>3</sup> Faculty of Information Technologies, L.N. Gumilyov Eurasian National University, 2 Satpayev Str., Astana 010008, Kazakhstan; yelibayeva\_gk@enu.kz
- \* Correspondence: m.milosz@pollub.pl; Tel.: +48-601-838-980

**Abstract:** This paper describes a method and technology for processing natural language texts and extracting data from the text that correspond to the semantics of an ontological model. The proposed method is distinguished by the use of a Large Language Model algorithm for text analysis. The extracted data are stored in an intermediate format, after which individuals and properties that reflect the specified semantics are programmatically created in the ontology. The proposed technology is implemented using the example of an ontological model that describes the geographical configuration and administrative–territorial division of Kazakhstan. The proposed method and technology can be applied in any subject areas for which ontological models have been developed. The results of the study can significantly improve the efficiency of using knowledge bases based on semantic networks by converting texts in natural languages into semantically linked data.

**Keywords:** ontology; Semantic Web; natural language processing; ChatGPT; Large Language Model; geographic question answering system



**Citation:** Mukanova, A.; Milosz, M.; Dauletkaliyeva, A.; Nazyrova, A.; Yelibayeva, G.; Kuzin, D.; Kussepova, L. LLM-Powered Natural Language Text Processing for Ontology Enrichment. *Appl. Sci.* **2024**, *14*, 5860. <https://doi.org/10.3390/app14135860>

Academic Editor: Elza Bontempi

Received: 15 June 2024

Revised: 1 July 2024

Accepted: 2 July 2024

Published: 4 July 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A large portion of the information accumulated and utilized by humanity is not machine-readable, meaning that it cannot be directly interpreted (understood) by a computer program. This includes textual, graphical, audiovisual, and other types of information. Undoubtedly, there are tools that facilitate human interaction with such information. These include various means of indexing, searching, transforming, and even recognizing patterns. However, all of these tools do not allow for the interpretation of information—only a human being is capable of doing so.

Machine-readable information includes all structured formats. This primarily includes information stored in relational and non-relational databases, as well as documents represented in XML format with semantic markup. A special place among the ways of representing machine-readable information is occupied by semantic technologies or the Semantic Web—a set of technologies for representing and using conceptualized information in electronic form. The foundation of the Semantic Web is the so-called “ontological” approach, which is based on representing any information in the form of a semantic graph of arbitrary structure, with concepts as nodes and relationships as edges.

Ontological modeling and the Semantic Web represent the next step in the evolution of machine-readable information representation methods. They provide the capability to implement conceptual models in electronic form, the logic of which is akin to human

reasoning. In addition to storing information, semantic models enable logical inference based on rules.

The ontological approach, based on Semantic Web technology, offers well-known advantages in designing data models for information systems:

1. An ontological model can be easily expanded and supplemented, including integration with models from other domains. This allows it to evolve as necessary.
2. An ontological model can be developed and used collaboratively by different organizations and expert groups. The use of namespaces allows for the division of development into parts corresponding to different knowledge domains.

The primary global idea of the Semantic Web is to transform the web into a global database with a semantic structure, making the data understandable for humans, as well as machine-readable. However, the ontological approach is also applied to solve specific problems in various domains:

1. The construction of knowledge bases with the capability for logical inference.
2. Semantic search.
3. The publication of linked open datasets.

Given that a large portion of the global information corpus is stored in an unstructured format, a critical task is the creation of effective methods for converting this information into a structured, machine-readable form. Assuming that the Semantic Web represents the most promising model for representing machine-readable information, it is necessary to develop methods for populating ontologies with information extracted from natural language text.

To extract data from text, various techniques and methods can be employed. Information extraction (IE) systems play a crucial role in identifying and extracting meaningful data from unstructured and semi-structured data sources [1]. These systems aim to identify a predefined set of concepts within a specific domain, while filtering out irrelevant information [1]. Web data extraction systems are particularly valuable in this context, as they enable the extraction of data from web sources, including HTML web pages, and can encompass elements within the page, as well as the full text of the page itself [2]. Furthermore, the process of data extraction from web sources has led to the development of various techniques, such as automatic web news extraction using tree edit distance [3], web data extraction based on partial tree alignment [4], and intelligent self-repairable web wrappers [5]. The application of all of these methods requires the implementation of specialized algorithms for each language individually and, when necessary, consideration of the subject area, which makes the process of information extraction quite labor-intensive and inefficient.

The most modern method of processing natural language texts is the use of neural network-based deep learning models, known as Large Language Models (LLMs). In this paper, the authors propose the use of the ChatGPT 3.5 model by OpenAI API (Application Programming Interface). The advantage of using LLMs for text analysis is their universality and flexibility, which eliminates the need to formalize grammatical rules or develop separate algorithms for each language.

Practical experience has demonstrated that generative ChatGPT models are quite proficient at extracting information from text and are capable of logical reasoning based on the assertions contained within the text. Based on these assertions, the model can form a data block in a formal manner in accordance with the instructions of the input prompt.

The output of the generative model can be ambiguous, meaning that the same input data can yield different results. To achieve a stable outcome of the desired quality, it is necessary to conduct a series of experiments with quality assessment.

The article consists of six sections (apart from the Introduction). In the following sections, we present the results of the literature review, the formulated research questions, and the architecture of the IT system used in the experiment. Then, we give a description of the experiment and its results. In the final part of the article, we present a discussion (in relation to the research questions) and conclusions.

## 2. Literature Review

The literature review is focused on the areas indicated in the Introduction section, namely natural language processing technologies, Large Language Models, the use of generative artificial intelligence and tools, and geographical information systems. The review used bibliographic and full-text databases such as Scopus, IEEE Xplore, Web of Science, SpringerLink, Science Direct, etc.

Data mining involves the process of extracting valuable patterns and information from large datasets [6]. It is a knowledge-discovery process that aims to find trends and regularities within corporate data warehouses [7]. On the other hand, text mining is a sub-speciality of knowledge discovery from data (KDD) and is focused on extracting useful information from massive amounts of unstructured or semi-structured text data [8]. Text mining differs from data mining in that it processes natural unstructured text, which is intended for human consumption, rather than structured databases designed for programmatic processing [9].

Natural language processing (NLP) is a critical component of text mining, enabling the analysis of unstructured text data. NLP has evolved over time, and its sub-problems are well-documented in the field [10]. Researchers have refined NLP tools for real-world applications, including speech-to-speech translation engines, sentiment analysis, and mining social media for information [11]. Furthermore, the use of pre-trained models for NLP has been the subject of recent surveys, highlighting the advancements in this area [12].

In the context of healthcare, text mining has been applied to electronic health records to extract symptoms and patient information [13]. This application has revealed challenges, such as the need for improved reporting of patient demographic characteristics [13]. Additionally, text mining has been used in the analysis of biomedical literature, where the volume of published research has been increasing rapidly, necessitating advanced text mining techniques for knowledge extraction [14].

The process of converting text to ontology involves mapping free text to concepts in an ontology, which has been explored in various domains, such as biomedicine, computational biology, and computer science [15–18]. This mapping is achieved through entity linking or annotation, which involves associating text with concepts in a knowledge graph or ontology [19,20]. The process typically includes identifying appropriate text spans in narratives and then mapping these text spans to target concepts in an ontology [20]. Named-entity normalization is also utilized to build a mapping relationship between named entities in text and ontology [21]. Furthermore, the use of semantic annotators has been highlighted in expanding users' queries with concepts and terms from vocabularies/ontologies, as well as in classifying retrieved documents based on their content or specific topics [17]. Additionally, the process involves the generation of relevant ontology terms, their definitions, and the relationships between them [22].

The generation of domain-specific ontologies from unstructured text corpus has been emphasized, demonstrating the need for domain-independent ontology-generation methods [23]. Furthermore, the use of standard representations, such as ontologies, has been highlighted to provide a unified view of information extracted from data [24]. The process also involves linking text to semantically similar classes in an ontology, which has been achieved through exemplar-based algorithms [25]. Moreover, the detection of appropriate frames from input text has been shown to improve the design quality of resulting ontologies, as frames can be directly mapped to ontology design patterns [26].

To extract data for filling an ontology, various techniques and systems have been developed. These include Text-to-Onto, which utilizes statistical methods like generalized association rule discovery and symbolic methods such as lexico-syntactic pattern method [27]. Smart-dog and OntoPrima are systems that extract data from technical data sheets and text to populate ontologies using NLP techniques and domain expert validation [28,29].

The integration of NLP with geospatial analytics represents a significant advancement in extracting and interpreting geospatial information from unstructured data. With the evolution of artificial intelligence, increased availability of digital text data, and enhanced computational power, NLP methods have become increasingly sophisticated, enabling the

identification of events, places, entities, and spatiotemporal patterns within geographic phenomena [30].

Exploring the capabilities of LLMs in geospatial data comprehension reveals that, beyond size, sophistication in model design is crucial for effectively synthesizing geospatial knowledge from textual information. Through innovative experimental approaches, such as probing for geo-coordinates and assessing geospatial reasoning with multidimensional scaling, insights into LLMs' geospatial awareness and reasoning abilities are uncovered. These findings highlight the potential and limitations of LLMs in facilitating informed geospatial decision-making, underscoring the need for more advanced developments in the field [31].

ChatGPT can be used for classifying domain terms according to upper ontologies [32]. This section delves into the use of ChatGPT as a tool for extracting data to build an ontology, providing valuable insights into the practical implementation of language models in ontology development. Another interesting use of ChatGPT is the automatic generation of SPARQL queries for ontologies [33].

GeoQA (Geographic Question Answering) [34] represents a burgeoning research domain within Geographic Information Science (GIScience), focusing on responding to geographic inquiries using natural language. Despite its promise, seamlessly merging structured geospatial data with unstructured natural language queries poses considerable challenges. Recent strides in LLMs have paved the way for natural language processing in diverse applications. This study introduces GeoQAMap, a novel system that translates natural language questions into SPARQL queries, retrieves geospatial data from Wikidata, and generates interactive maps as visual answers. Notably, GeoQAMap demonstrates potential for integration with additional geospatial datasets like OpenStreetMap and CityGML, facilitating complex geographic question answering involving advanced spatial operations.

In the context of GeoQA systems, a comprehensive framework is proposed that integrates natural language processing, machine learning, ontological reasoning, and geographic information systems (GIS). By leveraging GIS functionalities and spatial ontologies, the system demonstrates enhanced capabilities in accurately answering diverse geographic questions. Through empirical evaluation, it achieves an impressive overall accuracy of 90%, underscoring the importance of spatial reasoning and GIS in advancing GeoQA systems [35].

Enhanced YAGO2geo integrates OpenStreetMap data with YAGO2 to facilitate spatial reasoning for place-related queries. By translating natural language into logical representations and generating executable GeoSPARQL queries dynamically, it significantly improves question answering based on linked knowledge bases, as demonstrated on the Geospatial Gold Standard dataset [36].

Currently known advancements in the field of geospatial data analysis have been significantly enriched by the integration of LLMs like ChatGPT; LLMs are explored for their potential to interact with spatial databases using natural language. There is great interest in this innovative approach, as it presents a framework that could potentially transform the way geospatial data are analyzed by simplifying the generation and interpretation of geospatial SQL queries. Such developments are seen as pivotal in enhancing the accessibility and efficiency of complex geospatial analytics, marking a noteworthy progression in the field [37].

The advancement of GeoQA systems offers the promise of making geographic information accessible without GIS expertise, by answering natural language questions. Overcoming the challenge of geo-analytical queries, which demand GIS analysis for solutions, involves a novel question-parsing approach that leverages core spatial concepts and their roles in context-free grammar for question interpretation. This methodology allows for the transformation of geo-analytical questions into abstract GIS workflow expressions, enhancing the systems' capability to process and analyze complex geographic inquiries [38].

To sum up, the analysis of the literature indicates a lack of research on the possibilities of effective use of native artificial intelligence systems and LLM algorithms to develop ontological models in the area of geographical information systems.

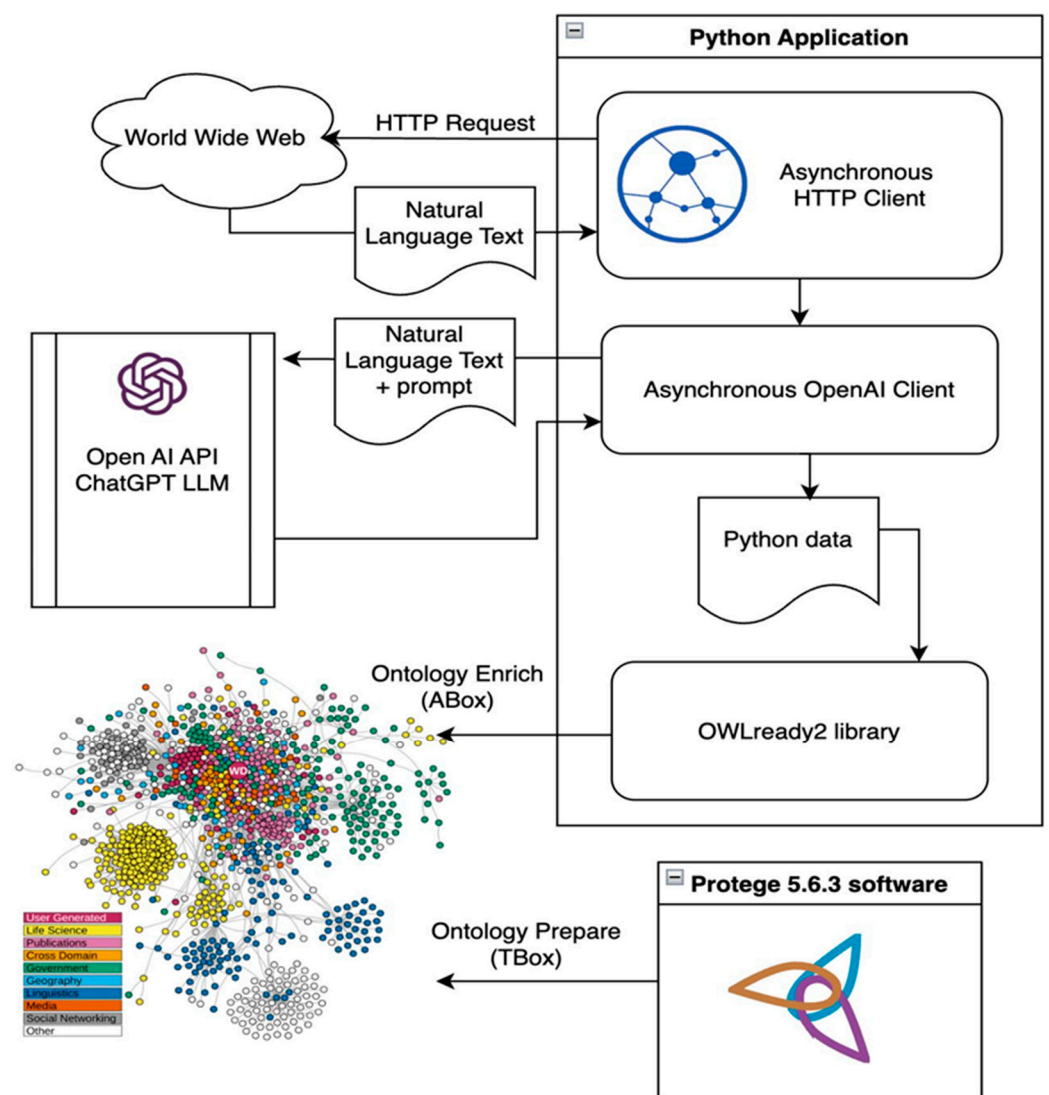
The introductory section's discussion and the findings from the literature review have culminated in the formulation of the three subsequent research questions:



- Q1. Is it feasible to obtain a machine-readable output from processing natural language texts using ChatGPT when the source texts are unadopted web pages?
- Q2. How consistent is the response from the OpenAI API when the query format and source text remain unchanged?
- Q3. Can the result from the OpenAI API query be automatically processed and the derived data be uploaded into an ontological model?

### 3. Architecture of the Information System for the Experiment

To obtain answers to the aforementioned research questions, the authors have designed the architecture for an information system that implements the stated tasks (Figure 1). The preliminary preparation of the ontology is carried out using the Protégé ontology editor, where a hierarchy of classes is created, and their object properties and data properties are defined. An example of the developed ontological model is provided in Section 4.4.



**Figure 1.** Architecture of the information system for data extraction.

The principal component of the system is a Python 3.9 application, which ensures the sequential execution of the following functions:

1. Making HTTP requests to predefined URLs in accordance with the subject domain of the ontology being developed.
2. Preparing a query based on the content of the web page obtained in the previous step and the description of the intermediate data representation format.
3. Executing a query to the OpenAI API GPT-3.5 and receiving data in the intermediate format (Python data).
4. Creating objects and defining their properties in the ontological model using the methods of the OWLready2 library.

The implementation of these functions is described in the following section.

#### 4. Description of the Experiment

##### 4.1. Retrieving Data from the WWW via HTTP Protocol

Web-page addresses for data extraction are prepared in advance by domain experts for the ontology being developed. An important advantage of the proposed technology is the separation of expertise between the domain expert and the knowledge engineer, who is responsible for representing knowledge in the form of an ontological model.

Data retrieval from the web is facilitated by the aiohttp library, which enables the asynchronous execution of HTTP requests. The result of a request is a complete web page containing a large amount of service data and markup elements not related to the domain. To reduce the volume of irrelevant data for subsequent processing, it is advisable to isolate the substantive parts of the page.

As an example, let us consider working with a Wikipedia page as a data source for an ontological geographic information system. In the code example shown in Figure 2, the BeautifulSoup parsing library is used to extract the title and the element containing the main content of the page. The code in Figure 2 parses a specific page (i.e., [https://en.wikipedia.org/wiki/Geography\\_of\\_Kazakhstan](https://en.wikipedia.org/wiki/Geography_of_Kazakhstan) (accessed on 10 June 2024)); for any other, it must be modified.

```
async def get_html(url):
    async with ClientSession() as session:
        async with session.get(url=url) as response:
            page = await response.read()
            soup = BeautifulSoup(page, "html.parser")
            title = soup.find('span', class_='mw-page-title-main').contents[0]
            body = soup.find('div', class_='mw-content-ltr mw-parser-output')
            return str(title) + "\n" + str(body)

prompt = await get_html('https://en.wikipedia.org/wiki/Geography_of_Kazakhstan')
```

**Figure 2.** Parsing wikipedia.org page.

##### 4.2. Preparing a Query for LLM ChatGPT 3.5

The input query for the generative model ChatGPT is formulated in natural language and can include formalized data. The aim of using ChatGPT is to obtain domain-specific data and present them in a specific machine-readable format. Thus, the query (prompt) should include the following information:

1. Web-page code.
2. A general description of the data to be extracted.
3. A description of the data presentation format.

Using Python as the data representation format is convenient. This will simplify their further use in the information system. It is important to note that the operation of the ChatGPT model is probabilistic, meaning that the result obtained in different iterations may vary, even with the same input query.

Through a series of experiments, the authors developed the following query format, which provides a stable result (the web-page code is not shown due to its large volume)—Figure 3.

```
Select mentioned in the text above countries, regions, settlements,
geographical objects such as lakes, rivers, mountains and return it in
three languages (Kazakh, English, Russian) as lists according to this
example:
countries = [
    ["name_in_english", [("name_in_kz", "kz"), ("name_in_en", "en"),
    ("name_in_ru", "ru")]],]
regions = [
    ["name_in_english", [("name_in_kz", "kz"), ("name_in_en", "en"),
    ("name_in_ru", "ru")]],]
settlements = [
    ["name_in_english", [("name_kz", "kz"), ("name_en", "en"),
    ("name_ru", "ru")]],]
objects = [
    ["name_in_english", [("name_kz", "kz"), ("name_en", "en"),
    ("name_ru", "ru")]],]
Add the affiliation of regions to countries in the format:
countries_regions = [(region_name_in_english,
country_name_in_english),]
Add the affiliation of settlements to regions in the format:
regions_settlements = [(settlement_name_in_english,
region_name_in_english),]
Add the affiliation of geographical objects to regions in the format:
regions_objects = [(object_name_in_english,
region_name_in_english),]
```

**Figure 3.** A request to ChatGPT to extract data from a web page into Python lists.

In this request, a textual description of the desired outcome is provided, along with the format for eight lists in Python format. The first five lists will contain individuals of classes—country, region, settlement, and object, while the remaining three lists will contain information about the affiliation of the region to the country, the settlement to the region, and the object to the region.

#### 4.3. Executing a Query to the OpenAI API and Receiving Data

To interact with the OpenAI API, the asynchronous library AsyncOpenAI is used. Figure 4 shows the program code, where an object of the class openAIClient is utilized to execute the query. The text of the above-mentioned query is contained in the file query.txt.

The response received from ChatGPT-3.5, corresponding to the above query (abbreviated), is shown in Figure 5. It is presented in Python code. Such a data representation format is convenient for further processing, as it does not require additional transformation or parsing and can be directly incorporated into the program handler.

```

async def analyze_wiki(openAIClient, prompt, query):
    completion = await openAIClient.chat.completions.create(model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": (str(prompt) + query)}])
    return completion.choices[0].message.content

async def main():
    prompt = await get_html('https://en.wikipedia.org/wiki/Geography_of_Kazakhstan')
    query = open('query.txt', mode='r').read()
    openAIClient = AsyncOpenAI(api_key=os.environ['OPENAI_API_KEY'])
    completion = await analyze_wiki(openAIClient, prompt, query)
    data = open('geo.py', mode='w')
    print(completion)
    data.write(completion)

```

**Figure 4.** Executing a query to the OpenAI API.

```

countries = [
    ["Kazakhstan", [("Қазақстан", "kz"), ("Kazakhstan", "en"), ("Казахстан", "ru")]],
]

regions = [
    ["Almaty Region", [("Алматы облысы", "kz"), ("Almaty Region", "en")]],
    ["Akmoła Region", [("Ақмола облысы", "kz"), ("Akmoła Region", "en")]],
    ["Aktobe Region", [("Ақтөбе облысы", "kz"), ("Aktobe Region", "en")]],
]

settlements = [
    ["Almaty", [("Алматы", "kz"), ("Almaty", "en"), ("Алматы", "ru")]],
    ["Astana", [("Астана", "kz"), ("Astana", "en"), ("Астана", "ru")]],
    ["Kostanay", [("Қостанай", "kz"), ("Kostanay", "en"), ("Костанай", "ru")]],
    ["Pavlodar", [("Павлодар", "kz"), ("Pavlodar", "en"), ("Павлодар", "ru")]],
]

objects = [
    ["Altai Mountains", [("Алтай таулары", "kz"), ("Altai Mountains", "en")]],
    ["Caspian Sea", [("Каспий теңізі", "kz"), ("Caspian Sea", "en")]],
    ["Zhetysu District", [("Жетісу ауданы", "kz"), ("Zhetysu District", "en")]],
]

countries_regions = [
    ("Almaty Region", "Kazakhstan"),
    ("Akmoła Region", "Kazakhstan"),
    ("Aktobe Region", "Kazakhstan"),
]

regions_settlements = [
    ("Almaty", "Almaty Region"),
    ("Kostanay", "Kostanay Region"),
    ("Pavlodar", "Pavlodar Region"),
]

regions_objects = [
    ("Altai Mountains", "East Kazakhstan Region"),
    ("Caspian Sea", "Mangystau Region"),
]

```

**Figure 5.** Fragment of intermediate data extracted from the page text.

#### 4.4. Enrichment of the Ontological Model with the Obtained Data

For the practical application of the proposed technology and to obtain answers to the research questions, the authors developed an ontological model of geography and administrative–territorial division of countries. The hierarchy of classes and properties of the developed model is shown in Figure 6. The graph of the semantic network, generated using the ProtégéVOWL plugin, is presented in Figure 7.

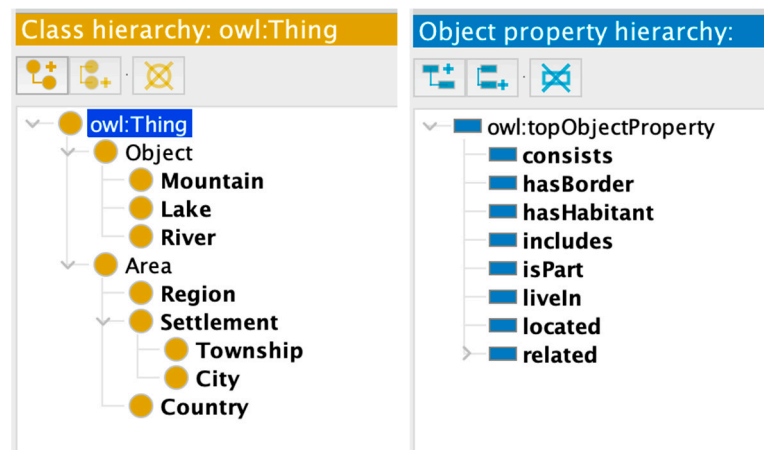


Figure 6. Hierarchy of classes and object properties of the ontological model.

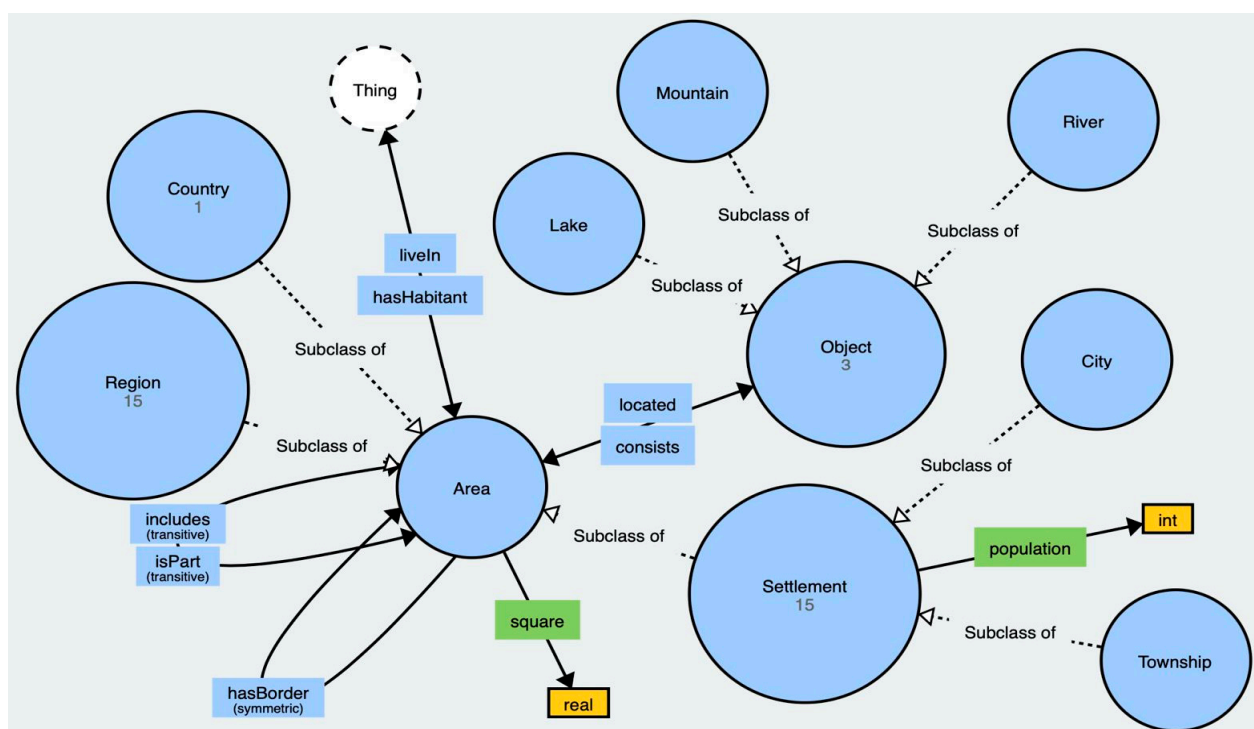


Figure 7. Graph of the semantic network of the geoinformation system.

At the next stage, the developed ontological model needs to be enriched with data extracted from the web-page text and presented in Python. This is implemented using program code, a fragment of which is shown in Figure 8. Interaction with the ontology occurs through classes of the OWLready2 library. For each individual, label properties are created in three languages.

After executing the program code, all individuals and their properties described in the data will be created in the ontology. The result of opening the created ontology in the Protégé editor is shown in Figure 9. The created ontology is fully consistent. The reasoner initializes without errors.



```

# Adding countries with multilanguage labels to ontology
for country in countries:
    print(country)
    country_individual = onto.Country(country[0])
    country_individual.label = []
    for label in country[1]:
        country_individual.label.append(locstr(label[0], lang=label[1]))
# Adding regions with multilanguage labels to ontology
for region in regions:
    print(region)
    region_individual = onto.Region(region[0])
    region_individual.label = []
    for label in region[1]:
        region_individual.label.append(locstr(label[0], lang=label[1]))
# Adding properties that adds region into country in ontology
for country_region in countries_regions:
    country_individual = onto.Country(country_region[1])
    region_individual = onto.Region(country_region[0])
    region_individual.isPart = [country_individual]

```

Figure 8. Fragment of the program code for creating objects in the ontology.

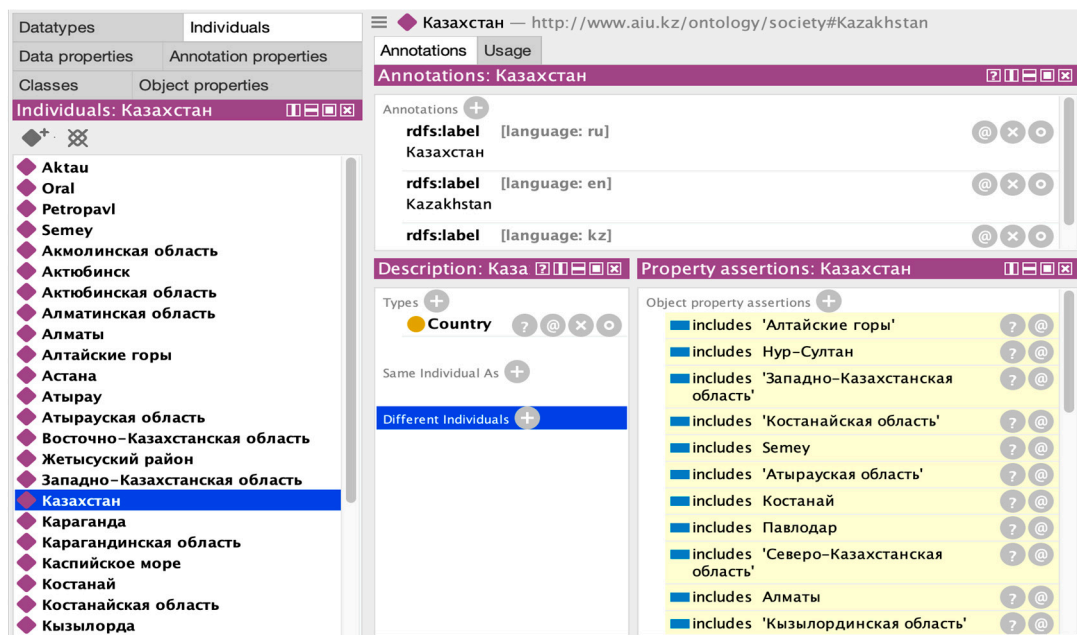


Figure 9. Created ontology in the Protégé editor with an initialized reasoner.

## 5. Results and Discussion

To assess the quality, completeness, and execution time of the task of extracting data from the text of a web page using ChatGPT 3.5, a series of experiments were conducted, during which the following data extraction results and their representation in a machine-readable format were collected (Table 1):

1. The total number of data elements found and included in the result (countries; regions; settlements; geographical features; and the relationships region–country, settlement–region, and object–region).

2. The number of irrelevant data elements included in the result, i.e., objects or facts not present in the text and considered “information noise” by the neural network.
3. The error rate in processing the page (the ratio of the number of irrelevant elements to the total number of elements).
4. The time taken to process the page.

**Table 1.** Results of experiments on extracting data from a web page using ChatGPT 3.5.

Experiment Number	Number of Elements Found	Quantity of Non-Relevant Elements	Error Rate	Processing Time, Seconds
1	18	1	5.56%	19
2	36	3	8.33%	22
3	36	5	13.89%	21
4	18	1	5.56%	20
5	42	7	16.67%	23
6	25	0	0.00%	18
7	16	0	0.00%	17

The obtained results indicate that a text analysis of a web page using ChatGPT is, as expected, much faster than an analysis performed by humans. However, the results of text processing can vary significantly even when the query is identical and the input data are the same. This variability is due to the nature of the algorithms used in the LLM model, which are stochastic in character. Additionally, the complexity and volume of the context itself also affect this parameter. In the conducted experiments, this is particularly important since the entire web page is included in the prompt, including markup tags that create “information noise”.

During the experiment, it was established that the number of data elements found on the page could vary from 18 to 42 in different experiments. The issue of data completeness can be addressed by making several identical queries and then combining the data.

A more serious problem is the retrieval of irrelevant elements in the query results—those for which information is completely absent on the processed web page. Such elements are generated by the model based on its own training datasets, and their identification requires additional verification. One way to perform such verification could be to check all found data elements for their presence in the page text. This is precisely how irrelevant data elements were identified during the execution of the above-described experiment.

The conducted research made it possible to answer the research questions posed.

- Q1. Is it feasible to obtain a machine-readable output from processing natural language texts using ChatGPT when the source texts are unadapted web pages? The answer to this research question: Yes, this possibility is confirmed by the experiments conducted and described in this work.
- Q2. How consistent is the response from the OpenAI API when the query format and source text remain unchanged? The answer to this research question: The experiments conducted showed that the results obtained using the same query can randomly differ from one another, and there is no reliable way to avoid this.
- Q3. Can the result from the OpenAI API query be automatically processed and the derived data be uploaded into an ontological model? The answer to this research question: Yes, it is possible, and such automatic processing was implemented in the program code during the research.

## 6. Conclusions

In the article, a method and technology for processing natural language texts and extracting data corresponding to the semantics of an ontological model are described. This method is characterized by the use of a Large Language Model algorithm for text analysis. The data extracted are saved in an intermediate format, followed by the programmatic creation of individuals and properties in the ontology. The technology is implemented using

the ontological model that describes the geographical configuration and administrative–territorial division of Kazakhstan as an example. This method and technology can be applied in various subject areas for which ontological models have been developed.

The architecture of an information system and Python program code for the automatic processing of responses to OpenAI API queries and integration of data into the ontological model are also developed.

A series of experiments were conducted to assess the quality, completeness, and time efficiency of the task of extracting data from web-page text using ChatGPT 3.5. The experiments showed that query results can vary even with identical input data. In some cases, the data may contain irrelevant elements, the appearance of which is a consequence of the stochastic nature of the applied machine learning algorithms. This indicates the need for additional procedures to control the obtained results.

Overall, the study demonstrated that using the ChatGPT 3.5 model allows for the effective extraction of information from unadapted web pages and the loading of data into an ontological model, which opens up broad prospects for automating text processing in various subject areas.

The developed software in Python enables the automation of the process of processing the results of queries to the OpenAI API and the integration of the obtained data into an ontological model, significantly simplifying the process of creating and updating semantic knowledge bases. It is important to note that the potential ambiguity of results must be taken into account, and additional research should be conducted to ensure the quality and stability of the data obtained.

**Author Contributions:** Conceptualization, A.M. and M.M.; methodology, A.N. and D.K.; software, A.M., L.K. and A.D.; validation, M.M., A.D. and A.M.; formal analysis, G.Y.; investigation, A.N. and D.K.; resources, A.M.; data curation, A.M.; writing—original draft preparation, A.D. and L.K.; writing—review and editing, M.M.; visualization, A.N. and A.M.; supervision, M.M.; project administration, A.M.; funding acquisition, A.M. and M.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP19577922).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ranjan, R.; Vathsala, H.; Koolagudi, S.G. Profile generation from web sources: An information extraction system. *Soc. Netw. Anal. Min.* **2022**, *12*, 2. [\[CrossRef\]](#)
2. Jayasankar, U.; Thirumal, V.; Ponnurangam, D. A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 119–140. [\[CrossRef\]](#)
3. Dey, R.; Balabantaray, R.C.; Mohanty, S. Sliding window based off-line handwritten text recognition using edit distance. *Multimed. Tools Appl.* **2022**, *81*, 22761–22788. [\[CrossRef\]](#)
4. Rupapara, V.; Narra, M.; Gonda, N.K.; Thipparthi, K. Relevant data node extraction: A web data extraction method for non contagious data. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 500–505. [\[CrossRef\]](#)
5. Xu, T.; Feng, A.; Song, X.; Gao, Z.; Zeng, X. Chinese News Data Extraction System Based on Readability Algorithm. In Proceedings of the 6th International Conference on Artificial Intelligence and Security, Hohhot, China, 17–20 July 2020; Springer: Singapore, 2020; pp. 153–164. [\[CrossRef\]](#)
6. Plotnikova, V.; Dumas, M.; Milani, F. Adaptations of data mining methodologies: A systematic literature review. *PeerJ Comput. Sci.* **2020**, *6*, e267. [\[CrossRef\]](#) [\[PubMed\]](#)

7. Verma, A.; Bhattacharya, P.; Bodkhe, U.; Ladha, A.; Tanwar, S. Dams: Dynamic association for view materialization based on rule mining scheme. In Proceedings of the 3rd International Conference on Recent Innovations in Computing, Jammu, India, 20–21 March 2020; Springer: Singapore, 2020; pp. 529–544. [\[CrossRef\]](#)
8. Fareri, S.; Fantoni, G.; Chiarello, F.; Coli, E.; Binda, A. Estimating Industry 4.0 impact on job profiles and skills using text mining. *Comput. Ind.* **2020**, *118*, 103222. [\[CrossRef\]](#)
9. Zong, C.; Xia, R.; Zhang, J. *Text Data Mining*; Springer: Singapore, 2021; Volume 711, p. 712. [\[CrossRef\]](#)
10. Chowdhary, K.; Chowdhary, K.R. Natural language processing. In *Fundamentals of Artificial Intelligence*; Springer: New Delhi, India, 2020; pp. 603–649. [\[CrossRef\]](#)
11. Torfi, A.; Shirvani, R.A.; Keneshloo, Y.; Tavaf, N.; Fox, E.A. Natural language processing advancements by deep learning: A survey. *arXiv* **2020**, arXiv:2003.01200. [\[CrossRef\]](#)
12. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **2020**, *63*, 1872–1897. [\[CrossRef\]](#)
13. Koleck, T.A.; Dreisbach, C.; Bourne, P.E.; Bakken, S. Natural language processing of symptoms documented in free-text narratives of electronic health records: A systematic review. *J. Am. Med. Inform. Assoc.* **2019**, *26*, 364–379. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Nadif, M.; Role, F. Unsupervised and self-supervised deep learning approaches for biomedical text mining. *Brief. Bioinform.* **2021**, *22*, 1592–1603. [\[CrossRef\]](#)
15. Demner-Fushman, D.; Elhadad, N.; Friedman, C. Natural language processing for health-related texts. In *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*; Springer International Publishing: Cham, Switzerland, 2021; pp. 241–272. [\[CrossRef\]](#)
16. Kersloot, M.G.; van Putten, F.J.; Abu-Hanna, A.; Cornet, R.; Arts, D.L. Natural language processing algorithms for mapping clinical text fragments onto ontology concepts: A systematic review and recommendations for future studies. *J. Biomed. Semant.* **2020**, *11*, 14. [\[CrossRef\]](#)
17. Tamine, L.; Goeuriot, L. Semantic information retrieval on medical texts: Research challenges, survey, and open issues. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 14. [\[CrossRef\]](#)
18. Li, Y.; Thomas, M.A.; Osei-Bryson, K.M. Ontology-based data mining model management for self-service knowledge discovery. *Inf. Syst. Front.* **2017**, *19*, 925–943. [\[CrossRef\]](#)
19. Prokhorov, V.; Pilehvar, M.T.; Collier, N. Generating knowledge graph paths from textual definitions using sequence-to-sequence models. *arXiv* **2019**, arXiv:1904.02996. [\[CrossRef\]](#)
20. Oommen, C.; Howlett-Prieto, Q.; Carrithers, M.D.; Hier, D.B. Inter-Rater Agreement for the Annotation of Neurologic Concepts in Electronic Health Records. *medRxiv* **2022**. [\[CrossRef\]](#)
21. Wang, Y.; Fan, X.; Chen, L.; Chang, E.I.; Ananiadou, S.; Tsujii, J.; Xu, Y. Mapping anatomical related entities to human body parts based on wikipedia in discharge summaries. *BMC Bioinform.* **2019**, *20*, 430. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Islam, N.; Syed, D.; Shaikh, Z.A. Semantic Web: An Overview and a. net-based Tool for Knowledge Extraction and Ontology Development. In *Semantic Technologies for Intelligent Industry 4.0 Applications*; River Publishers: New York, NY, USA, 2023; pp. 169–197. [\[CrossRef\]](#)
23. Elnagar, S.; Yoon, V.; Thomas, M.A. An automatic ontology generation framework with an organizational perspective. *arXiv* **2022**, arXiv:2201.05910. [\[CrossRef\]](#)
24. Pezoulas, V.C.; Sakellarios, A.; Kleber, M.; Bosch, J.A.; Van der Laan, S.W.; Lamers, F.; Lehtimäki, T.; Marz, W.; Fotiadis, D.I. A hybrid data harmonization workflow using word embeddings for the interlinking of heterogeneous cross-domain clinical data structures. In Proceedings of the 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), Virtual Conference, 27–30 July 2021; pp. 1–4. [\[CrossRef\]](#)
25. Ghoniem, R.M.; Alhelwa, N.; Shaalan, K. A novel hybrid genetic-whale optimization model for ontology learning from Arabic text. *Algorithms* **2019**, *12*, 182. [\[CrossRef\]](#)
26. Liu, K.; Chen, Y.; Liu, J.; Zuo, X.; Zhao, J. Extracting events and their relations from texts: A survey on recent research progress and challenges. *AI Open* **2020**, *1*, 22–39. [\[CrossRef\]](#)
27. Houssein, E.H.; Mohamed, R.E.; Ali, A.A. Machine learning techniques for biomedical natural language processing: A comprehensive review. *IEEE Access* **2021**, *9*, 140628–140653. [\[CrossRef\]](#)
28. González, L.; García-Barriocanal, E.; Sicilia, M.A. Entity linking as a population mechanism for skill ontologies: Evaluating the use of ESCO and Wikidata. In Proceedings of the Metadata and Semantic Research: 14th International Conference, MTSR 2020, Madrid, Spain, 2–4 December 2020; Revised Selected Papers 14. Springer International Publishing: Cham, Switzerland, 2021; pp. 116–122. [\[CrossRef\]](#)
29. Melo, D.; Rodrigues, I.P.; Varagnolo, D. A strategy for archives metadata representation on CIDOC-CRM and knowledge discovery. *Semant. Web* **2023**, *14*, 553–584. [\[CrossRef\]](#)
30. Zhang, C.; Zhang, C.; Zheng, S.; Qiao, Y.; Li, C.; Zhang, M.; Dam, S.K.; Thwal, C.M.; Tun, Y.L.; Huy, L.L. A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need? *arXiv* **2023**, arXiv:2303.11717Jiang.
31. Bhandari, P.; Anastasopoulos, A.; Pfoser, D. Are large language models geospatially knowledgeable? In Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems, Hamburg, Germany, 13–16 November 2023; pp. 1–4. [\[CrossRef\]](#)

32. Rodrigues, F.H.; Lopes, A.G.; dos Santos, N.O.; Garcia, L.F.; Carbonera, J.L.; Abel, M. On the Use of ChatGPT for Classifying Domain Terms According to Upper Ontologies. In Proceedings of the 42nd International Conference on Conceptual Modeling, Lisbon, Portugal, 6–9 November 2023; Springer: Cham, Switzerland, 2023; pp. 249–258. [\[CrossRef\]](#)
33. Ekuobase, G.O.; Ebietomere, E.P. Latest Applications of Semantic Web Technologies for Service Industry. In *Semantic Web Technologies*; CRC Press: Boca Raton, FL, USA, 2022; pp. 73–104. [\[CrossRef\]](#)
34. Feng, Y.; Ding, L.; Xiao, G. GeoQAMap-Geographic Question Answering with Maps Leveraging LLM and Open Knowledge Base (Short Paper). In Proceedings of the 12th International Conference on Geographic Information Science (GIScience 2023), Leeds, UK, 12–15 September 2023. [\[CrossRef\]](#)
35. Scheider, S.; Nyamsuren, E.; Krüger, H.; Xu, H. Geo-analytical question-answering with GIS. *Int. J. Digit. Earth* **2021**, *14*, 1–14. [\[CrossRef\]](#)
36. Yang, J.; Jang, H.; Yu, K. Geographic Knowledge Base Question Answering over OpenStreetMap. *ISPRS Int. J. Geo-Inf.* **2023**, *13*, 10. [\[CrossRef\]](#)
37. Jiang, Y.; Yang, C. Is ChatGPT a Good Geospatial Data Analyst? Exploring the Integration of Natural Language into Structured Query Language within a Spatial Database. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 26. [\[CrossRef\]](#)
38. Xu, H.; Nyamsuren, E.; Scheider, S.; Top, E. A grammar for interpreting geo-analytical questions as concept transformations. *Int. J. Geogr. Inf. Sci.* **2023**, *37*, 276–306. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.