From Text to Knowledge: Leveraging LLMs and RAG for Relationship Extraction in Ontologies and Thesauri*

Antonios Georgakopoulos[1,*,†], Jacco van Ossenbruggen[2,†] and Lise Stork[1,†]

[1] *Informatics Institute, University of Amsterdam, 1098 XH Amsterdam, The Netherlands*
[2] *Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands*

Abstract

Ontologies, vocabularies, and thesauri provide a shared conceptualisation for a domain. Manually maintaining and updating such knowledge systems when knowledge changes, does not scale for large domains, such as in biomedicine. Recently, large language models (LLMs) have been increasingly used as tools in knowledge engineering processes, offering new possibilities for the automatic creation and maintenance of knowledge systems. This work explores how LLMs can be leveraged for the automated extension of such knowledge systems. Specifically, we build on the DRAGON-AI framework, which integrates Retrieval-Augmented Generation (RAG) to provide LLMs with access to external knowledge sources for more faithful outputs. We investigate the ability of the framework to predict relationships between a given knowledge system and a novel concept. We do so for both an ontology and a thesaurus, and analyse the impact of (i) enriching prompts with contextual information as well as more clear instructions, (ii) an alternative retrieval approach, and (iii) using a conversational model versus an instruction-following model. The results show superior quality in the ontology generations for all models and approaches compared to the thesaurus. The two models show varied performance across the different experiment configurations with only the conversational model showing notably improved performance, in terms of F1, for the ontology with the custom retrieval approach.

Keywords

Ontology, Thesaurus, Large Language Models, Knowledge Engineering, Prompting, RAG, DRAGON-AI

CEUR-WS.org/Vol-3967/ELMKE_2024_paper_4.pdf

1. Introduction

In the field of Artificial Intelligence (AI), ontologies [1] and thesauri are used to explain and represent formal knowledge for a specific domain. These structured representations can capture human knowledge in a way that computers can process and interpret. They depict the concepts and relations of a shared conceptualisation in a structural way [2]. Many information retrieval applications depend on the accuracy of these knowledge systems, since they contain domain knowledge which is vital for the correct and efficient functionality of these applications. The increasing complexity of intelligent systems renders the use of an up-to-date knowledge system imperative [3], however manually creating and updating such structures with the help of domain experts can be both time-consuming and costly [4]. Moreover, techniques for constructing such knowledge structures, such as ontologies, in an automatic manner that do not utilise a large language model (LLM), usually require the structure's schema to be predefined–a non-trivial task–as well as domain experts to process and evaluate the results [5, 6].

LLMs [7, 8] constitute the state-of-the-art in the NLP domain due to their advanced capabilities in language understanding [9]. Their integration in Knowledge Engineering (KE) workflows shows a promising direction in automating the construction and extension of knowledge-holding structures. A successful blend of LLMs and KE is evident in [10, 11], where authors use a language model to extract information from unstructured text and in combination with a domain-specific ontology, they were able to populate a knowledge graph in an automated fashion. In [12] the authors built an ontology

EKAW 2024: EKAW 2024 Workshops, Tutorials, Posters and Demos, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024), November 26-28, 2024, Amsterdam, The Netherlands.

*Corresponding author.
†These authors contributed equally.
a.georgakopoulos@uva.nl (A. Georgakopoulos); jacco.van.ossenbruggen@vu.nl (J. v. Ossenbruggen); l.stork@uva.nl (L. Stork)
0000-0002-7748-4715 (J. v. Ossenbruggen); 0000-0002-2146-4803 (L. Stork)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

by first feeding competency questions (CQs) in a language model, and then instructing the model to extract relevant concepts and relationships. By using the model's output suggestions, they are able to successfully integrate them into an ontology. Nevertheless, LLMs also suffer from hallucination problems [13, 14] which means that they show a tendency of creating their own fabricated content that is not in-line with the ground truth.

Retrieval-Augmented Generation (RAG) is an emerging technology that aims to reduce hallucinations in LLMs by including external knowledge into the prompts [15, 16]. The RAG architecture usually consists of an LLM with its own parametric memory, as well as an external data source (e.g. a vector database) that is supplementing the LLM with additional information in order to enrich the prompt with more relevant and accurate knowledge. This approach is known for its speed and cost-efficiency, making it a preferred approach for connecting an LLM to proprietary data and providing responses that are grounded to the data. [17] shows an example of an effective utilisation of the RAG approach to reduce hallucinations. By providing the LLM relevant information along with the user query, the authors are able to mitigate the phenomenon of hallucinations. Almanac [18] is a RAG-infused LLM framework that uses external tools such as search engines and medical databases. The evaluation of this approach on 130 clinical questions shows that it achieves superior performance on the factuality and accuracy of the responses, compared to an LLM that does not utilise the RAG architecture.

This work further explores the capabilities of LLMs in combination with the RAG to automatically construct knowledge systems (ontologies and thesauri) from textual data. We will leverage the DRAGON-AI framework [19], that utilises a RAG architecture, and evaluate its ability on relationship properties generation. Our approach will test and measure the completeness and accuracy of the generated relationship properties by exploring the effectiveness of various techniques within the DRAGON-AI framework. We will extend the framework by implementing a customised approach that could potentially yield better results than the already existing approach. We also measure the performance of the task of relationship generation on different large language models. We aim to understand the impact of RAG for the task of ontology and thesaurus extension, and specifically:

RQ.1 Ontologies versus Thesauri. *How effectively does a RAG system perform in the task of extending ontologies, with complex, heterogeneous schema, versus extending thesauri with predefined, simpler schema?*

RQ.2 LLM-type. *Which type of LLM (conversational or instruction-following) is more effective in the tasks of ontology and thesaurus extension via RAG?*

RQ.3 Prompting and Retrieval. *How do variations in retrieval algorithms and prompt structures impact the effectiveness of RAG for ontology and thesaurus extension?*

The code for reproducibility of the experiments can be accessed through our GitHub repository[1].

2. Related Work

Automatic Thesaurus Generation Although available research on creating thesauri in an automated manner is scarce, we can identify some specific approaches that offer a promising guiding principle. In [20] the authors use parallel corpora to create a bilingual thesaurus. By leveraging multiple methodologies, such as morphological analysis, part-of-speech (POS) tagging, and statistical weighting, they are able to generate a large number of thesaurus entries. Despite this fruitful endeavour to create the bilingual thesaurus, a major problem arises when words or phrases in one language do not have a direct equivalent in the other language. More advanced techniques could help alleviate this issue due to their ability to capture the relationships between languages in a more optimal way. [21] proposes a statistical method that incorporates syntactic parsing along with word co-occurrence to understand the relationships between the word in a large number of medical abstracts. Although the thesaurus

[1] https://github.com/Antonis-Georgakopoulos/curate-gpt

produced from this approach contains an adequate number of entries, more advanced techniques could help in discovering more complex semantic relationships between words that usually demand a deeper understanding of the words' meaning and the context.

Rule-based and Statistical OL The Ontology Learning (OL) field tries to implement a variety of different techniques and approaches for automatically creating an ontology from text [22]. It accomplishes that by interpreting the intent and context behind data and not just processing it as raw information. Before the use of deep neural networks, the field of OL heavily relied on the more traditional machine learning (ML) techniques that include statistical and rule-based methods [23, 24, 25]. In [26] the authors utilised data mining techniques and heuristic-based approaches to generate an ontology from domain-specific text. The approach mentioned in [27] utilised POS taggers as well as syntactic parsers to expand ontologies by parsing unstructured text. Various works incorporate the identification of lexico-syntactic patterns as part of the pipeline for effective OL implementations [28, 23]. The Text2Onto framework [29] is able to perform the task of automatic ontology creation by performing NLP techniques to identify taxonomies and other linguistic classifications from text. Although these approaches provide an easy and transparent way to construct an ontology, they lack the ability to generalize their performance to unseen data patterns and they are time-consuming due to their dependency on human intervention [30].

Deep Learning-based OL Deep learning (DL) approaches have shown improved performance on specific NLP tasks compared to the more traditional ML approaches [31]. These methods are more capable of creating word embeddings, understand the dependencies between words in a longer sequence of text and extract concepts and relationships in a more efficient way. A plethora of academic literature appears to utilise deep neural networks for the task of Entity Recognition (ER) in order to extract specific entities from the unstructured text [32, 33]. In [34] the authors implemented a DL algorithm based on an unsupervised neural network architecture in order to classify the taxonomic relationships in the ontology. A combination of Gated Recurrent Units (GRU) and Long Short Term Memory (LSTM) was utilised in [35] for the task of Named Entity Recognition (NER) in order to extract the ontological concepts from text. Another hybrid model was proposed by [36] where they fused a Bidirectional LSTM with a Recurrent Neural Network (RNN) so that they could analyse the input text data in both directions. Despite the benefits that these approaches offer over the more traditional ML techniques, they require a high level of expertise for the training of deep learning models and may encounter difficulties understanding more domain-specific terminology [37].

LLM-based OL Due to the novelty of this area, existing research about the use of LLMs for automatically extending ontologies or thesauri is limited. One study explores zero-shot prompting for ontology extension across diverse knowledge domains and found that, while LLMs show potential, they still require task-specific fine-tuning for more practical use, as it significantly improves performance across all tasks [38]. In [6], the authors extract hierarchical concepts, based on a given query concept, by prompting the LLM to return relevant subconcepts. Even though results show promise, hallucinations occur, polluting generation results. To address these issues, the DRAGON-AI framework [19] explores the impact of RAG for ontology generation, aiming to minimising hallucinations. The authors test their approach on the task of ontology term completion. By providing a small free-text definition of a novel concept, their approach aims to automatically extend the ontology with that concept. Overall, the quality of the AI-generated ontology definitions was inferior compared to those constructed by human experts, showing that human expertise is often still crucial for validation. Our approach employs the DRAGON-AI framework to understand better how such models deal with the complexity of ontologies versus thesauri, the effect of a novel retrieval approach, and the impact of different LLM types, specifically those trained to follow instructions, versus those trained for conversation.

3. Problem Definition

Ontology extension This work explores the problem of extending an ontology with novel classes. An ontology $\mathcal{O}$ is a formal representation of knowledge within a domain, typically defined as a tuple: $\mathcal{O} = (C, R, I, P, A)$ where C is the set of concepts (or classes) representing entities in the domain, R is the set of relations between these concepts, I is the set of individuals (or instances) representing specific entities, P is the set of properties (or attributes) that describe characteristics of the concepts and individuals, and A is the set of axioms that enforce logical constraints and define relationships between concepts, individuals, and properties.

The concept of ontology extension in this paper refers to the enrichment of an existing ontology $\mathcal{O}$: given a novel unseen concept or query term $t \in C'$, we predict target relations $r_t \in R$ and target concepts $c_t \in C$ that relate the query term c_q to $\mathcal{O}$. Thus, the task is to predict (r_t, c_t), given c_q. In this work, for thesauri the task differs only for R, which consists of a fixed set of relationships: hierarchical and associative relations.

4. Overview of the RAG architecture

To answer our RQs, we employ the DRAGON-AI framework, which is based on the CURATE-GPT[2] project. The RAG architecture consists of three main components: a generative model (an LLM), a retriever, and a vector database (see Figure 1). The vector database contains extracted embeddings from the primary and secondary data sources, while the retriever performs a nearest-neighbour similarity search to retrieve the most relevant documents from the vector database. The generative model is responsible for predicting, given query term $c_q \in C'$, target relations $r_t \in R$ and target concepts $c_t \in C$.

The retriever will extract data from the vector database in two different phases depending on the methodological approach that is going to be used. During the first phase, which is mandatory across all approaches in our research, the retriever will gather examples from the primary data source used, that is essential for providing to the LLM a comprehensive understanding of the knowledge system's schema. The second phase, which is optional, involves retrieving examples from secondary data sources that can be used to further improve the quality of the generated LLM responses. We refer to these secondary data sources with B (background knowledge).

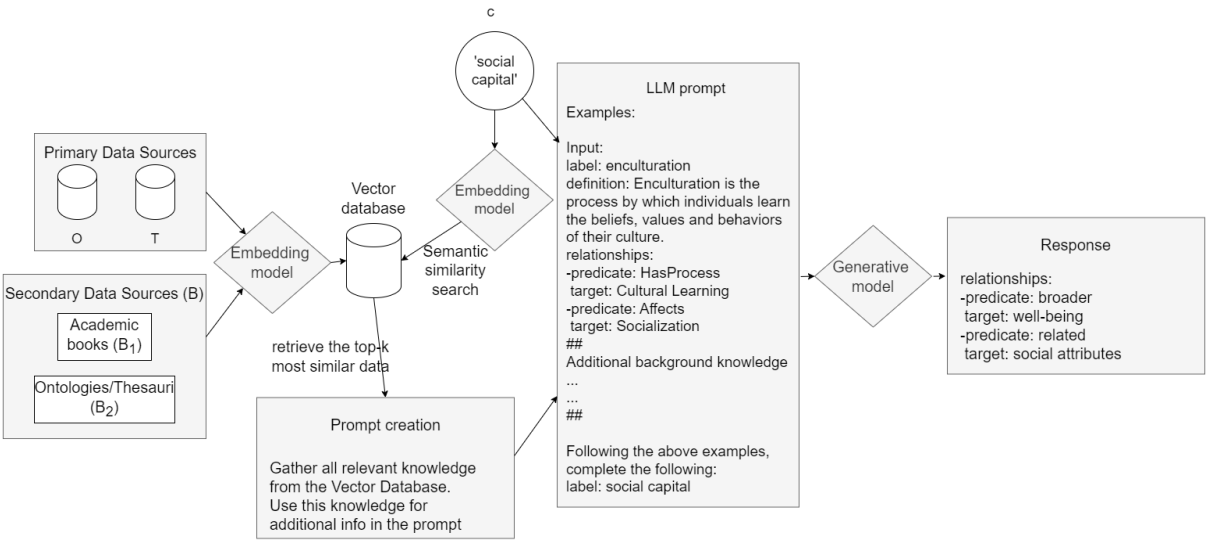

Figure 1: Overview of the RAG architecture

[2]https://github.com/monarch-initiative/curate-gpt

Table 1
Overview of all the combinations of entities, approaches and generative models used in this work.

Entity	Approach	Generative model
Thesaurus	DRAGON-AI-NB	GPT-3.5-TURBO
		GPT-3.5-TURBO-INSTRUCT
	DRAGON-AI	GPT-3.5-TURBO
		GPT-3.5-TURBO-INSTRUCT
	DRAGON-AI-CUSTOM	GPT-3.5-TURBO
		GPT-3.5-TURBO-INSTRUCT
Ontology	DRAGON-AI-NB	GPT-3.5-TURBO
		GPT-3.5-TURBO-INSTRUCT
	DRAGON-AI	GPT-3.5-TURBO
		GPT-3.5-TURBO-INSTRUCT
	DRAGON-AI-CUSTOM	GPT-3.5-TURBO
		GPT-3.5-TURBO-INSTRUCT

5. Experimental Methodology

5.1. Experimental Setup

To adequately address the RQs presented in this paper, we will conduct multiple experiments with various combinations of the knowledge systems, generative models, and methodological approaches. A detailed summary of the combinations can be seen in Table 1. For each knowledge system (thesaurus and ontology, **RQ.1**) we will employ two different LLMs (**RQ.2**) and for each such combination we will test the effect of three different methodological approaches (**RQ.3**). The following sections describe the different parts that synthesise the final methodological and architectural approach of this work. In Section 5.2 we outline the distinct characteristics of the two main data sources used in this work, while in Section 5.3 we present the different LLMs that generate the relations. Section 5.4 details the different strategies used to evaluate the quality of the generated relations.

5.2. Knowledge Systems

To ensure insightful results from the experiments and explore **RQ.1**, we reviewed various ontologies and thesauri to identify those that contained a sufficient diversity of relationships. Our choices are detailed below.

Thesaurus The first data source for our experiments is the ELSST (European Language Social Science Thesaurus) [39]. The ELSST thesaurus is a multilingual thesaurus for the social sciences, developed by CESSDA and its national service providers. ELSST covers core social science aspects such as politics, sociology, economics, and education, and contains 3422 concepts in total. ELSST contains the following relationships:

1. `broader`: Indicates the concept that is more general than the current term. For the *central government* entity the broader concept is *government*.
2. `narrower`: Specifies the scope of the current term and provides a subcategory. For the *central government* entities such as *coalition government* and *minority government* are narrower concepts.
3. `related`: This is an entity that is related to the current term in a non-hierarchical manner. For example, *bureaucracy* is a concept related to the *central government*.

In Table 2 we can see the number of occurrences for each relationship property in the ELLST thesaurus.

Table 2
Number of occurrences (N) for each predicate in the ELSST thesaurus.

Predicate	N
related	5668
broader	3533
narrower	3533

Ontology The second data source is an ontology, namely the BioAssay Ontology (BAO) [40]. The BAO ontology contains 8043 concepts and was chosen due to its plethora of diverse relationships and concepts. The BAO ontology contains the descriptions of various chemical biology experiments and their results. Table 3 shows the number of occurrences for each distinct relationship in the BAO ontology.

Table 3
Number of occurrences (N) for each predicate in the BAO ontology

Predicate	N
subClassOf	8949
HasRole	247
HasDetectionMethod	201
HasEndpoint	145
HasBioassayType	139
subPropertyOf	110
HasAssayFormat	101
HasSubstrate	95
HasAssayDesignMethod	83
HasOrganism	78
owl:inverseOf	48
HasCellLine	43
UsesDetectionInstrument	29
HasAssaySupportingMethod	28
range	27
HasExperimentalSetting	27
HasMeasuredEntity	26
HasParticipant	13
HasDetectedEntity	13
InvolvesBiologicalProcess	11
HasAssayMethod	7
HasPart	5

Predicate	N
domain	4
IsTransfectedInto	4
HasModeOfAction	3
Stains	2
IsRegulatedBy	2
HasQuality	2
InvolvesMolecularFunction	2
HasPreparationMethod	1
IsEndpointOf	1
HasFunction	1
Detects	1
HasAssayKitComponent	1
owl:equivalentClass	1
Reports	1
DetectsPhenotype	1
IsBindingSiteOf	1
IsRegulatorOf	1
HasBindingSite	1
Encodes	1
HasAssociatedDisease	1
DerivesFrom	1

5.3. Large Language Models

To effectively address **RQ.2**, we have chosen to compare the GPT-3.5-TURBO with the GPT-3.5-TURBO-INSTRUCT LLMs, developed by OpenAI. The GPT-3.5-TURBO model [41] is optimised for a variety of natural language understanding tasks such as machine translation and natural language inference. As a result, this kind of model is well suited for tackling problems that involve text generation.

The GPT-3.5-TURBO-INSTRUCT model [3] is a specific variation of the GPT-3.5-TURBO model that is trained to follow instructions. This model uses techniques, such as in-context learning, to understand a given instruction. The GPT-3.5-TURBO-INSTRUCT model appears promising for our work, as our task depends on specific instructions being followed correctly.

[3] https://platform.openai.com/docs/models/gpt-3-5-turbo

5.4. Prompting and Retrieval

To address **RQ.3**, we will run three different approaches based on varying retrieval and prompting approaches. Below, we discuss the prompt engineering and retrieval variations, after which we detail how these are employed in the three distinct approaches.

Prompt Engineering LLMs show a tendency to hallucinate responses, as mentioned in Section 1. To tackle this issue and optimise the performance of the tested LLMs, it is essential that we provide the models with a well-constructed prompt that meets the specific requirements for our task. Although the CURATE-GPT framework already uses a prompt that instructs an LLM to generate content according to a set of examples, we aim at optimising the prompt for ontology extension task. As a result, we deemed it necessary to construct a new prompt that clearly specifies the requirements for generating relationships without over-extending the context of the prompt. The new prompt can be found in our GitHub repository[4].

Retrieval Method Continuing with the exploration of techniques that could potentially enhance the quality of the generated outputs, we decided to implement a different retrieval methodology. The DRAGON-AI framework uses the Maximal Marginal Relevance (MMR) algorithm [42] to retrieve text from the vector database, balancing diversity and relevance of the retrieved results, thus reducing redundancy of the results. While the methodology offers accurate results, we believe that a more dedicated approach would improve the outcomes of our task. To accurately generate the relationship properties for a query term, the LLMs have to understand the connections between all terms in the structure. To enrich the prompt with relevant background knowledge B containing information about the query term and other relevant terms, the secondary data sources should be searched with a query term that is a combination of the main query term and the retrieved examples. These retrieved examples are the most relevant terms to our main query term and we believe that these will form the majority of the relationships. For example, if the main query term is *famine* and the retrieved example entities include: *hunger*, *infant feeding*, *forged migration* etc., then we will generate query pairs such as: *famine hunger*, *famine infant feeding*, *famine forged migration*, etc. By querying the secondary data sources with these word combinations, we hypothesise that the discovery of parts of text that contain both terms increases. The models can then infer the relationship between these terms according to the context of the passage and their own parametric memory.

Methodological Approaches The methodological approaches that we are going to follow are:

1. DRAGON-AI-NB approach: This approach does not utilise the background knowledge B part of the RAG architecture for retrieving additional resources. It only provides the LLM with 10 examples from the tested ontology or thesaurus that are semantically similar to the query term. The plain DRAGON-AI-NB approach will be used as a baseline.

2. DRAGON-AI approach: This approach includes the full DRAGON-AI approach, including supplementary background knowledge B which augments the prompt of the LLM with information relevant to the query term. The retrieval algorithm used is based on the pre-existing CURATE-GPT implementation.

3. DRAGON-AI-CUSTOM approach: This approach customises the DRAGON-AI approach by adapting the methodology for retrieving examples from B as well as the prompting technique, following the adaptations described above.

[4] https://github.com/Antonis-Georgakopoulos/curate-gpt/blob/main/assets/custom_prompt.txt

5.5. Evaluation Metrics

For all generated relationships of all the query terms in the test set (test set creation is described in Section 5.6), we will calculate the true positives, false positives and false negatives, which we define in the following way:

True positive : given c_q, the predicted target tuple (r_t, c_t) matches one from the test set.

False negative : given c_q, the predicted relationship tuple (r_t, c_t) does not exist in the test set.

False positive : none of the tuples (r, c_t) for c_q from the test set were predicted.

These measurements can help us calculate several important evaluation metrics that can shed light on the overall performance of the models in the task of ontology and thesaurus extension. These evaluation metrics include precision, recall and F1 score. We opt for F1 over accuracy, due to the imbalanced nature of the ontology and thesaurus.

Apart from the evaluation mentioned above, we also follow two different approaches for evaluating the generated relationships of an entity: a strict approach and a lenient approach. Both these approaches penalise incorrect predictions in a different way.

Lenient approach generated relationships that do not exactly match the gold standard relationships but are valid by inference (via hierarchical or subsumption relationships) are not counted as incorrect (0), but as partially correct (0.5). The inspiration of this evaluation method was drawn from the approach that was followed in [19].

Strict approach generated relationships that do not exactly match the gold standard relationships but are valid by inference are counted as incorrect (0).

5.6. Test Set Creation

We create a test set for both the ELSST thesaurus and the BAO ontology to test the different approaches. To ensure that there is no data leakage between the ontology $\mathcal{O}$ and the test set, we additionally perform a postprocessing step that we describe below.

Ontology and Thesaurus Partitioning For our experiments, we artificially create an ontology extension $\mathcal{O}'$. We do so by removing tuples (c_q, r_t, q_t) from the base ontology $\mathcal{O}$, to serve as our test set. For this, we analysed the distributions of both the ELSST thesaurus and the BAO ontology to make sure that the predicate distribution of the extension is similar to the distribution of the remaining base ontology. As can be seen in the Figure 2 below, these distributions do not follow a normal distribution pattern. Consequently, we can use stratified sampling to extract a valid C' set. Stratified sampling ensures that all subgroups withing the overall population are represented in the sample. By proportionally representing each subgroup, the test set accurately reflects the overall population of the initial dataset. Both the C' for the BAO ontology as well as for the ELSST thesaurus, contain 200 entities.

Data Leakage First, we removed any reference to concepts from C'. For each of the query terms c_q contained in $\mathcal{O}'$, we iterated through $\mathcal{O}$ and removed every mention of that term. We followed the same approach for the ontology and thesauri when used as secondary data sources.

6. Results

The following section presents results for the experiments (as outlined in Table 1). Both lenient and strict evaluation methods were carried out as mentioned in Section 5.5. After conducting our experiments,

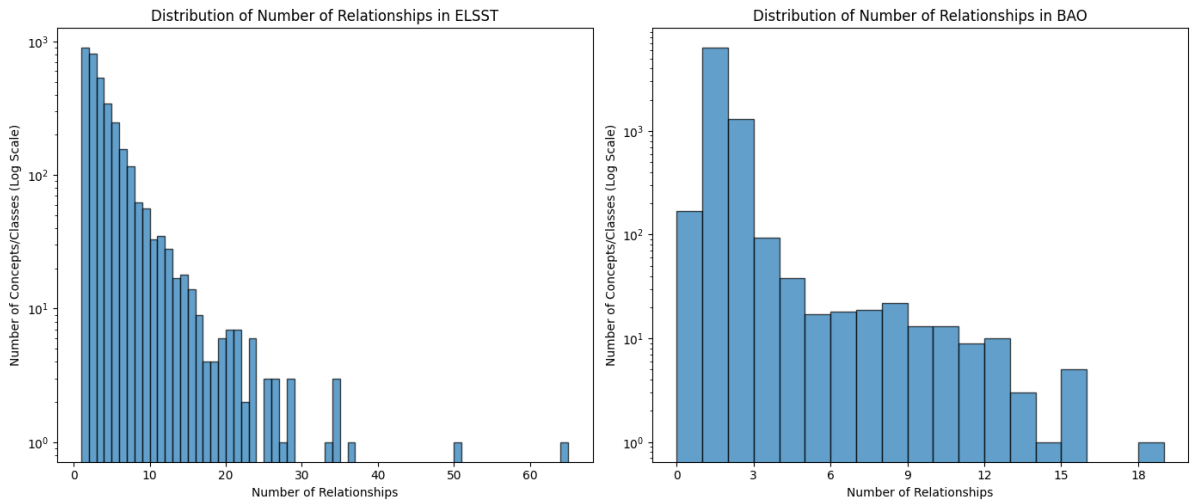

Figure 2: Overall distribution of Relationships for the ELSST thesaurus and the BAO ontology.

Table 4
True positives (TP), false positives (FP), false negatives (FN), recall (R.), precision (P.), F1 score and number of generated relationships (N) for the GPT-3.5-TURBO versus the GPT-3.5-TURBO-CONSTRUCT LLMs per approach for the ELSST.

	GPT-3.5-TURBO						
	TP	FP	FN	R.	P.	F1	N
DRAGON-AI-NB	55	307	691	0.073	0.151	0.099	362
DRAGON-AI	39	578	707	0.052	0.063	0.057	617
DRAGON-AI-CUSTOM	58	508	688	0.077	0.102	0.088	566

	GPT-3.5-TURBO-INSTRUCT						
	TP	FP	FN	R.	P.	F1	N
DRAGON-AI-NB	44	312	702	0.058	0.123	0.079	359
DRAGON-AI	53	614	703.5	0.07	0.079	0.077	674
DRAGON-AI-CUSTOM	55	734	691	0.073	0.069	0.071	792

we did not find a significant difference between the two evaluation methods and therefore decided to include only results for the strict evaluation. Results of the lenient evaluation can be found in our GitHub repository[1].

6.1. ELSST

6.1.1. Results for the ELSST Thesaurus

Table 4 provides a summary of the results for the ELSST thesaurus. We additionally provide the number of total generated relationships for each approach. To answer **RQ.2** by comparing the performance that each model exhibits in generating relationships for the ELSST. From the table 4, we can observe that for the GPT-3.5-TURBO model the DRAGON-AI-NB approach achieves a moderate number of true positives and false negatives, resulting in a more balanced F1 score. In contrast, for the same approach, the instruction-following model yields fewer true positives and higher false positives, leading to worse overall performance for the same approach. A similar pattern can be observed for the DRAGON-AI-CUSTOM approach as well, where every evaluation metric for the conversational language model shows a higher value. For the DRAGON-AI approach the results demonstrate an opposite scenario, where the GPT-3.5-TURBO-INSTRUCT model performs better across every evaluation setting. The results on table 4 along with the graphical representation of the outcomes in Figure 3, suggest that the GPT-3.5-TURBO model generally achieves higher scores in different configuration approaches compared to the GPT-3.5-TURBO-INSTRUCT model.

To address **RQ.3**, we will compare the performances of the different methodological approaches for each LLM. Starting with the conversational model, the DRAGON-AI-NB approach appears to be more balanced compared to the other two approaches. This methodological approach also yields a higher precision score, which logically follows from the fact that the approach generates less overall relationships and as a result we have fewer false positive cases. A related trend can be observed for the

GPT-3.5-TURBO-INSTRUCT model where the DRAGON-AI-NB approach shows higher F1 score and precision values. However, the recall score is comparatively lower than that of the other two approaches, showing a limitation in predicting the gold standard label data. For predicting the maximum number of true positive cases, the DRAGON-AI-CUSTOM approach is the most effective for both models, due to the large number of relationships that it generates. Overall, the DRAGON-AI-NB approach appears to be the most promising methodology for every LLM for the ELSST.

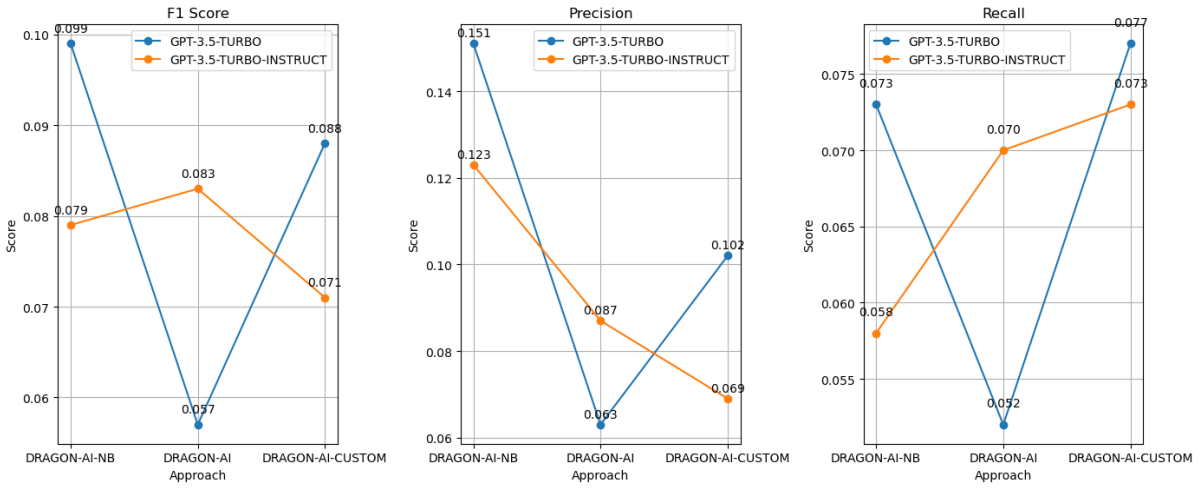

Figure 3: Comparison of the different approaches for each evaluation metric for the ELSST

6.1.2. Performance Comparison by Predicate Type

By looking at Figure 4 it is evident that both models perform the worst, looking at the `narrower` predicate due to the low precision, recall and F1-score values. This means that the models are unable to correctly predict most of the golden standard (c_q, r_t, c_t) tuples where r_t = `narrower`. For test examples where r_t = `related` there is great variability in the metrics for all three approaches with the DRAGON-AI-CUSTOM approach appearing to be the most balanced. The `broader` predicate demonstrates the most harmonious performance due to the higher F1 score across all methodological approaches.

Comparing the results from the two models, the `broader` predicate appears to be the most accurately generated and thus more easily comprehended by the two models, resulting in greater accuracy when linking concepts with that specific predicate. When we observe the `related` predicate we can detect that it almost always produces the lowest precision scores in every methodology. This could be attributed to the fact that the majority of the generated predicates belong to the `related` relationship type. If we examine a part of a prompt that we provide to the LLM, we can indeed observe that for the examples that were given to the model, the majority of the relationship types contained in the examples are of type `related`. As a result, the models could exhibit a bias towards generating this specific relationship type more than any other relationship type.

6.2. BAO

6.2.1. Results for the BAO Ontology

This section provides an analysis of the results obtained from our experiments performed on the BAO test set. To investigate **RQ.2** we examine the performance of each LLM on every methodological approach. Looking at table 5 as well as Figure 5, both the DRAGON-AI-NB and DRAGON-AI approaches show a similar pattern in the outcome of the prediction task with the F1 score and recall values for both approaches being greater for the instruction-following model. Both approaches return a higher amount of true positive cases compared to the conversational model. However, the GPT-3.5-TURBO

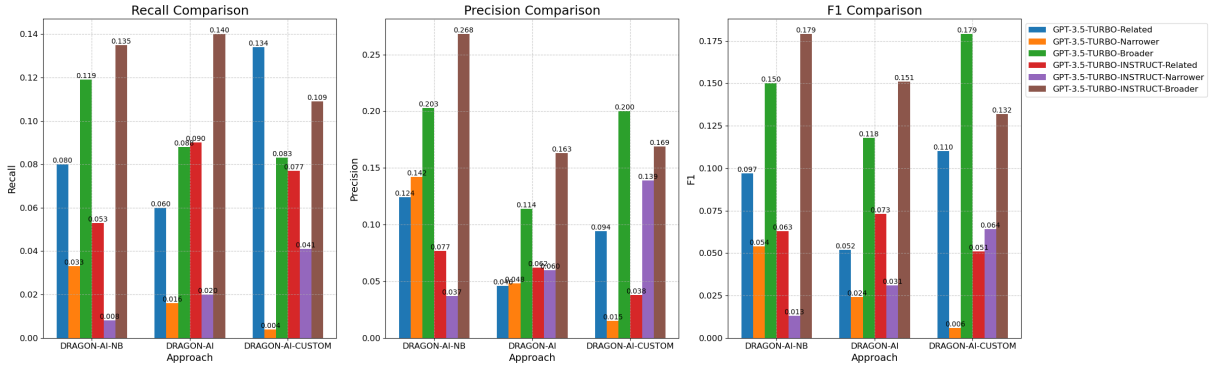

Figure 4: Comparison of selected evaluation metrics of each predicate type for both models

Table 5
True positives (TP), false positives (FP), false negatives (FN), recall (R.), precision (P.), F1 score and number of generated relationships (N) for the GPT-3.5-TURBO versus the GPT-3.5-TURBO-CONSTRUCT LLMs per approach for the BAO.

	GPT-3.5-TURBO						
	TP	FP	FN	R.	P.	F1	N
DRAGON-AI-NB	33	37	218	0.131	0.471	0.205	90
DRAGON-AI	37	66	214	0.147	0.359	0.209	120
DRAGON-AI-CUSTOM	77	158	174	0.306	0.327	0.316	246

	GPT-3.5-TURBO-INSTRUCT						
	TP	FP	FN	R.	P.	F1	N
DRAGON-AI-NB	52	99	199	0.207	0.344	0.258	178
DRAGON-AI	55	119	196	0.219	0.316	0.258	210
DRAGON-AI-CUSTOM	58	279	193	0.231	0.172	0.197	380

model exhibits a lower number of false positive cases, therefore managing to outperform the GPT-3.5-TURBO-INSTRUCT model in terms of the precision metric. Looking at the DRAGON-AI-CUSTOM approach, it is evident that the GPT-3.5-TURBO model yields a comparatively better performance than the GPT-3.5-TURBO-INSTRUCT model. The latter generally achieves more optimal performance when being supplied with a more minimal prompt, whereas GPT-3.5-TURBO performs best when the prompt is supplemented with additional information.

Shifting our attention to **RQ.3**, it is undeniably clear that for the GPT-3.5-TURBO model, the DRAGON-AI-CUSTOM approach achieves the best performance across every methodology used. Not only it manages to generate the most amount of true positive cases, but also it shows the lowest amount of false negative cases. This indicates that the utilisation of this approach was able to generate a big proportion of relationships that belong in the golden standard data and the methodology was perfectly complimented by the conversational model. For the instruction-following model, both DRAGON-AI-NB and DRAGON-AI approaches show a similar performance that is superior to that of the DRAGON-AI-CUSTOM approach.

6.2.2. Performance Comparison for the `subClassOf` Predicate

The analysis of the results for the predicate `subClassOf` can help us discover any potential hidden pattern in the way that models generate the relationships that contain this specific predicate. This stems from the fact that the `subClassOf` predicate is the most common relationship type in the BAO. As seen in the Figure 6, the result patterns for the `subClassOf` predicate appear to be almost identical to the ones for the overall test set. The combination of the GPT-3.5-TURBO model with the DRAGON-AI-CUSTOM approach appears to constitute the most balanced option with an F1 score that is a lot higher than any other approach. For both models the DRAGON-AI-CUSTOM approach appears to be generating comparatively the greatest proportion of the golden standard data, out of the three approaches, due to the higher recall value. Nevertheless, the DRAGON-AI-CUSTOM approach does not perform as well for the GPT-3.5-TURBO-INSTRUCT model, as was also observed in the evaluation of the overall test set.

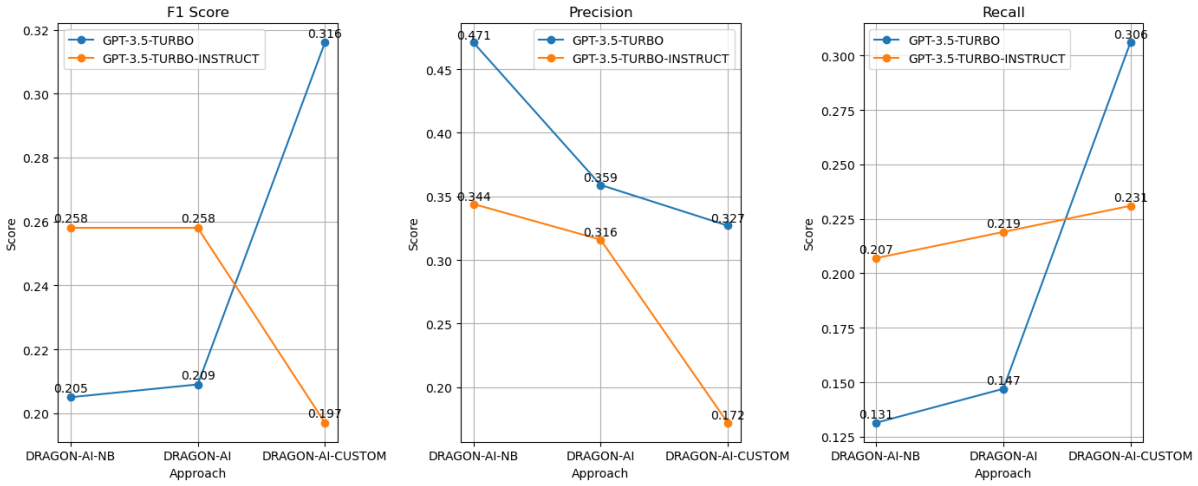

Figure 5: Comparison of the different approaches for each evaluation metric for the BAO ontology

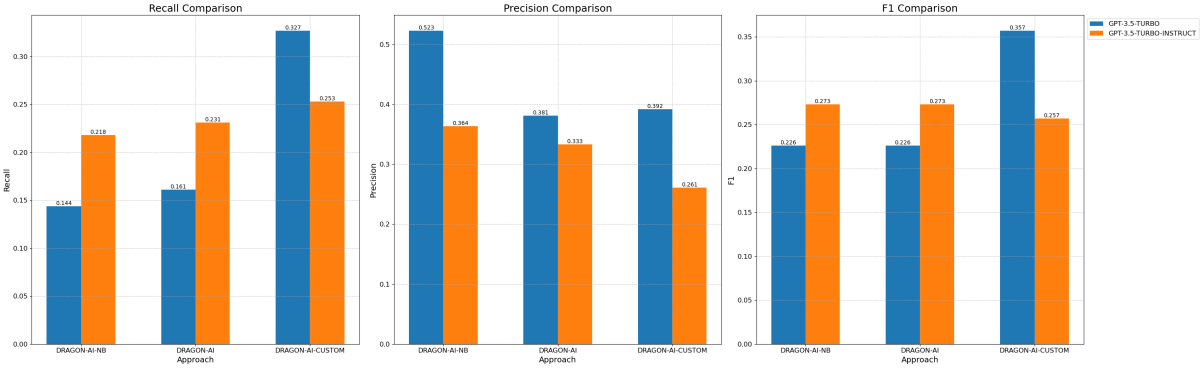

Figure 6: Comparison of selected evaluation metrics of the `subClassOf` predicate for both models

7. Discussion

In this section, we will reflect on the results that we obtained from our experiments and try to address each research question, providing an analysis of the possible answers.

RQ.1 Judging from the results that stem from the experiments that we conducted, we can observe that both LLMs that were tested seem to be performing better on the ontology extension task than the thesaurus extension task. This becomes evident when we notice the prominent difference between the F1 scores for the ontology and the thesaurus (regardless of the approach used). One inference that can be made from these results is that when we are testing a structure that contains a predicate that is significantly more prevalent than others, the models show a more advanced capability of predicting the golden standard relationships regarding this specific predicate. On the other hand, when the predicates are more evenly distributed in terms of their occurrences in the dataset, the models yield comparatively lower performance. As a result, it can be asserted that the models become slightly more biased in predicting the most common predicates.

It is important to note that for ontologies with greater variability and distribution, the results could be different. It is necessary to further evaluate the three approaches tested (DRAGON-AI-NB, DRAGON-AI and DRAGON-AI-CUSTOM) with additional knowledge structures.

RQ.2 Results for **RQ.2** are inconclusive. For the ELSST thesaurus, if we would like to have the most balanced approach, we would choose the GPT-3.5-TURBO model and utilise the DRAGON-AI-NB approach. In a case that we want to provide additional context into the prompt, it is evident that

the GPT-3.5-TURBO model with the DRAGON-AI-CUSTOM approach is the best combination. The GPT-3.5-TURBO-INSTRUCT appears to be beneficial only when we follow the DRAGON-AI approach.

For the BAO ontology, it is clear that the GPT-3.5-TURBO model combined with the DRAGON-AI-CUSTOM approach gives us the best overall performance. The same model should be chosen in a scenario where we would like to minimise the amount of false positive cases that the model generates. However, in a situation where there is not additional context available for enriching the prompt, then the GPT-3.5-TURBO-INSTRUCT model becomes a better overall choice.

We identify a weakness of the GPT-3.5-TURBO-INSTRUCT model when handling a plethora of diverse data. Although the DRAGON-AI-CUSTOM approach contains a prompt with specific steps for the model to follow, it appears that the GPT-3.5-TURBO-INSTRUCT model does not perform as well in understanding the task as well as separating the different sections of the prompt. On the other hand, the GPT-3.5-TURBO model seems to benefit more from the instructions given in the prompt and does not face difficulties regarding the additional context that we provide in the prompt. Thus, it becomes clear that if we would prefer to enrich the prompt with extra information, then the GPT-3.5-TURBO model is the better choice, while the GPT-3.5-TURBO-INSTRUCT model should be chosen when the prompt is relatively short but contains detailed instructions.

The results from [19] further validate our findings with respect to the ontology structures. In the paper the authors observed a weakness of the Nous-Hermes-13b model, which was fine-tuned over a plethora of instructions, to predict the relationships of various ontologies as accurately as the GPT-3.5-TURBO model.

RQ.3 The results suggest that for the GPT-3.5-TURBO model, the DRAGON-AI-CUSTOM approach always yields more accurate outcomes than the DRAGON-AI approach for both the ELSST and the BAO ontology. This means that a properly structured prompt with clear instructions and a more diverse context is beneficial for that specific model. Conversely, for the GPT-3.5-TURBO-INSTRUCT model, methodologies that use a more minimal prompt and less diverse context appear to be more overall balanced. To provide a more complete answer, one additional evaluation to perform would be to test both models with minimal additions for each experiments. For example, we could conduct experiments with more minimal prompts and only a single source of additional data (e.g. PDF files) to get a more comprehensive evaluation.

8. Conclusion

Enriching ontologies and thesauri with relationships that were generated from large language models is a challenging task and requires multiple resources and different models regarding the algorithmic approach that is being followed. We tested one ontology (BAO) and one thesaurus (ELSST) with two different LLMs and three distinct methodologies in order to understand the strengths and weaknesses of each approach and each model. We extended the functionalities of the CURATE-GPT framework in order to develop a customised approach of extracting relevant data from the vector database. As the additional data that enriches the prompts, we utilised various data sources so that we could introduce further diversity in the context of the prompt. Moreover, we enhanced the prompt of the CURATE-GPT framework with a more directive prompt that contains clear steps for the task of relationships generation.

The experiments yielded varied results, showing that there is no clear answer as to what model and approach performs best, as it depends highly on the specifications of each approach. Our customised approach in combination with one specific model appears to be the most beneficial for the generation of the relationship properties for the ontology. Studies such as these, that aim at getting a better understanding of the use of LLMs and RAG systems for knowledge engineering tasks, are important for better development and reuse of ontologies. Manually updating ontologies or thesauri does not scale for large domains (such as biomedicine), resulting in ontologies that are not up-to-date and limiting reuse.

Acknowledgements

This work is partially funded by the Netherlands Organisation of Scientific Research (NWO), ODISSEI Roadmap project: 184.035.014.

References

[1] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge acquisition 5 (1993) 199–220.
[2] T. R. Gruber, Toward principles for the design of ontologies used for knowledge sharing?, International journal of human-computer studies 43 (1995) 907–928.
[3] J. I. Olszewska, J. Bermejo-Alonso, R. Sanz, Special issue on ontologies and standards for intelligent systems: editorial, The Knowledge Engineering Review 37 (2022) e6. doi:10.1017/S0269888922000030.
[4] A. Memariani, M. Glauer, F. Neuhaus, T. Mossakowski, J. Hastings, Automated and explainable ontology extension based on deep learning: A case study in the chemical domain, arXiv preprint arXiv:2109.09202 (2021).
[5] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, H. M. Abbasi, A survey of ontology learning techniques and applications, Database 2018 (2018) bay101.
[6] M. Funk, S. Hosemann, J. C. Jung, C. Lutz, Towards ontology construction with language models, arXiv preprint arXiv:2309.09898 (2023).
[7] T. Brants, A. Popat, P. Xu, F. J. Och, J. Dean, Large language models in machine translation, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 858–867.
[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[9] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, arXiv preprint arXiv:2402.06196 (2024).
[10] S. Yu, T. Huang, M. Liu, Z. Wang, Bear: Revolutionizing service domain knowledge graph construction with llm, in: F. Monti, S. Rinderle-Ma, A. Ruiz Cortés, Z. Zheng, M. Mecella (Eds.), Service-Oriented Computing, Springer Nature Switzerland, Cham, 2023, pp. 339–346.
[11] L. Stork, R. L. Zijdeman, I. Tiddi, A. ten Teije, Enabling social demography research using semantic technologies, in: A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), The Semantic Web, Springer Nature Switzerland, Cham, 2024, pp. 199–216.
[12] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An llm supported approach to ontology and knowledge graph construction, arXiv preprint arXiv:2403.08345 (2024).
[13] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, et al., A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, arXiv preprint arXiv:2302.04023 (2023).
[14] K. Lee, O. Firat, A. Agarwal, C. Fannjiang, D. Sussillo, Hallucinations in neural machine translation, 2019. URL: https://openreview.net/forum?id=SkxJ-309FQ.
[15] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.
[16] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, E. Grave, Atlas: Few-shot learning with retrieval augmented language models, arXiv preprint arXiv:2208.03299 (2022).
[17] P. Béchard, O. M. Ayala, Reducing hallucination in structured outputs via retrieval-augmented generation, arXiv preprint arXiv:2404.08189 (2024).
[18] C. Zakka, R. Shad, A. Chaurasia, A. R. Dalal, J. L. Kim, M. Moor, R. Fong, C. Phillips, K. Alexander,

E. Ashley, et al., Almanac—retrieval-augmented language models for clinical medicine, NEJM AI 1 (2024) AIoa2300068.
[19] S. Toro, A. V. Anagnostopoulos, S. Bello, K. Blumberg, R. Cameron, L. Carmody, A. D. Diehl, D. Dooley, W. Duncan, P. Fey, et al., Dynamic retrieval augmented generation of ontologies using artificial intelligence (dragon-ai), arXiv preprint arXiv:2312.10904 (2023).
[20] K. Kageura, K. Tsuji, A. Aizawa, Automatic thesaurus generation through multiple filtering, in: COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics, 2000.
[21] G. Grefenstette, Automatic thesaurus generation from raw text using knowledge-poor techniques, in: Making sense of Words. Ninth Annual Conference of the UW Centre for the New OED and text Research, 1993.
[22] A. Konys, Knowledge repository of ontology learning tools from text, Procedia Computer Science 159 (2019) 1614–1628.
[23] F. Xu, D. Kurz, J. Piskorski, S. Schmeier, A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping, in: M. González Rodríguez, C. P. Suarez Araujo (Eds.), Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain, 2002, pp. 224–230. URL: http://www.lrec-conf.org/proceedings/lrec2002/pdf/351.pdf.
[24] M. Missikoff, R. Navigli, P. Velardi, The usable ontology: An environment for building and assessing a domain ontology, in: International semantic web conference, Springer, 2002, pp. 39–53.
[25] D. Lonsdale, Y. Ding, D. W. Embley, A. Melby, Peppering knowledge sources with salt: Boosting conceptual content for ontology generation, in: Proceedings of the AAAI Workshop on Semantic Web Meets Language Resources, Edmonton, Alberta, Canada, 2002.
[26] J.-u. Kietz, A. Maedche, R. Volz, A method for semi-automatic ontology acquisition from a corporate intranet, Proc of Workshop Ontologies and Text, co-located with EKAW'2000 (2000).
[27] C. Roux, D. Proux, F. Rechenmann, L. Julliard, An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions., in: ECAI Workshop on Ontology Learning, Citeseer, 2000.
[28] D. Moldovan, R. Girju, An interactive tool for the rapid development of knowledge bases., International Journal on Artificial Intelligence Tools 10 (2001) 65–86. doi:10.1142/S0218213001000428.
[29] P. Cimiano, J. Völker, Text2onto: A framework for ontology learning and data-driven change discovery, in: International conference on application of natural language to information systems, Springer, 2005, pp. 227–238.
[30] F. N. Al-Aswadi, H. Y. Chan, K. H. Gan, Automatic ontology construction from text: a review from shallow to deep learning trend, Artificial Intelligence Review 53 (2020) 3901–3928.
[31] J. Zhan, B. Dahal, Using deep learning for short text understanding, Journal of Big Data 4 (2017) 1–15.
[32] J. Santoso, E. I. Setiawan, C. N. Purwanto, E. M. Yuniarno, M. Hariadi, M. H. Purnomo, Named entity recognition for extracting concept in ontology building on indonesian language using end-to-end bidirectional long short term memory, Expert Systems with Applications 176 (2021) 114856.
[33] Q. H. Ngo, T. Kechadi, N.-A. Le-Khac, Domain specific entity recognition with semantic-based deep learning approach, IEEE Access 9 (2021) 152892–152902.
[34] L. Khan, F. Luo, Ontology construction for information selection, in: 14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings., 2002, pp. 122–127. doi:10.1109/TAI.2002.1180796.
[35] P. Manda, S. SayedAhmed, S. D. Mohanty, Automated ontology-based annotation of scientific literature using deep learning, in: Proceedings of the international workshop on semantic Big Data, 2020, pp. 1–6.
[36] C. Lyu, B. Chen, Y. Ren, D. Ji, Long short-term memory rnn for biomedical named entity recognition, BMC Bioinformatics 18 (2017). doi:10.1186/s12859-017-1868-5.
[37] R. Du, H. An, K. Wang, W. Liu, A short review for ontology learning from text: Stride from shallow

learning, deep learning to large language models trend, arXiv preprint arXiv:2404.14991 (2024).
[38] H. Babaei Giglou, J. D'Souza, S. Auer, Llms4ol: Large language models for ontology learning, in: International Semantic Web Conference, Springer, 2023, pp. 408–427.
[39] CESSDA, S. P. (2023), The european language social science thesaurus (elsst) (version 4), https://elsst.cessda.eu/, 2023.
[40] U. Visser, S. Abeyruwan, U. Vempati, R. P. Smith, V. Lemmon, S. C. Schürer, Bioassay ontology (bao): a semantic description of bioassays and high-throughput screening results, BMC bioinformatics 12 (2011) 1–16.
[41] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.
[42] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, 1998, pp. 335–336.