



HAL
open science

Semantic vs. LLM-based approach: A case study of KOnPoTe vs. Claude for ontology population from French advertisements

Aya Nour Elimane Sahbi, Céline Alec, Pierre Beust

► **To cite this version:**

Aya Nour Elimane Sahbi, Céline Alec, Pierre Beust. Semantic vs. LLM-based approach: A case study of KOnPoTe vs. Claude for ontology population from French advertisements. *Data and Knowledge Engineering*, 2025, 156, pp.102392. <10.1016/j.datak.2024.102392>. <hal-04823061>

HAL Id: hal-04823061

<https://hal.science/hal-04823061v1>

Submitted on 29 Sep 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License



Semantic vs. LLM-based approach: A case study of KOnPoTe vs. Claude for ontology population from French advertisements

Aya Sahbi ^{a,*}, Céline Alec ^a, Pierre Beust ^{a,b}

^a Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ, GREYC UMR 6072, F-14000 Caen, France

^b Univ Rennes, Inria, CNRS, IRISA - UMR 6074, F-35000 Rennes, France

ARTICLE INFO

Keywords:

Ontology population

LLM

Textual descriptions

ABSTRACT

Automatic ontology population is the process of identifying, extracting, and integrating relevant information from diverse sources to instantiate the classes and properties specified in an ontology, thereby creating a Knowledge Graph (KG) for a particular domain. In this study, we evaluate two approaches for ontology population from text: KOnPoTe, a semantic technique that employs textual and domain knowledge analysis, and a generative AI method leveraging Claude, a Large Language Model (LLM). We conduct comparative experiments on three French advertisement domains: real estate, boats, and restaurants to assess the performance of these techniques. Our analysis highlights the respective strengths and limitations of the semantic approach and the LLM-based one in the context of the ontology population process.

1. Introduction

The web is flooded with human-generated texts promoting products, services, or places. These descriptive advertisements often present information in incomplete sentences, designed more to catch attention than to inform comprehensively. Identifying, extracting, structuring, and integrating information from this specialized content into an ontology enables automatic ontology population to organize and effectively utilize this unstructured descriptive content. Ontologies organize information in a structured manner by linking entities through semantic relationships. Ontology population involves enriching an ontology by adding concrete instances. The resulting instantiated ontology can then be used as a knowledge graph, providing an efficient representation of knowledge that enables machines to reason and answer complex questions. This paper specifically addresses automatic ontology population from textual advertisements by comparing a semantic approach with a Large Language Model (LLM)-based approach. This work is an extended version of the papers by [1,2], providing additional details, new experiments, and more extensive analysis of the results.

The two main contributions of this paper are:

- A comparative analysis of two ontology population approaches: KOnPoTe, a semantic method developed by [3], and an LLM-based technique using the model Claude.¹ Our experiments focus on three application domain ontologies: Real Estate, Boats, and Restaurants. To the best of our knowledge, such a comparative analysis has not yet been presented within the semantic web community.

* Corresponding author at: Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ, GREYC UMR 6072, F-14000 Caen, France.

E-mail address: aya-nour-elimane.sahbi@unicaen.fr (A. Sahbi).

¹ <https://www.anthropic.com/index/introducing-claude>

<https://doi.org/10.1016/j.datak.2024.102392>

Received 15 June 2024; Received in revised form 25 November 2024; Accepted 27 November 2024

Available online 4 December 2024

0169-023X/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

- A novel study examining short and incomplete French texts. This case study presents unique challenges for the population process since most related work deals with full-length English text. Additionally, Claude has not been extensively evaluated on French data. Thus, this work presents a preliminary exploration of Claude’s multilingual comprehension capabilities for the French language through the use of French prompts and inputs.

This paper is organized as follows: Section 2 presents related work, Section 3 describes our case study, data, and the compared approaches, while Section 4 details the experiments and analyzes the results. Finally, the last section concludes and presents future work.

2. Related works

Ontology population refers to the process of adding instances to an ontology, either automatically or manually. Populating an ontology from text often involves Information Extraction (IE) tasks, such as Named Entity Recognition (NER), Open Information Extraction (OIE), Relation Extraction (RE), and event extraction, as presented by [4–7]. These techniques are widely studied within the Semantic Web community, as evidenced by recent works such as [8–10]. Extracting relevant information is not the only task to be performed; the population process itself requires the correct association of extracted instances with appropriate classes or properties, according to predefined ontological axioms. In this section, we explore two categories of approaches for populating ontologies from textual documents: semantic approaches and techniques based on Large Language Models (LLMs).

2.1. Ontology population from a semantic perspective

Several semantic works, citing [11–15], discuss various methods of ontology population. Some approaches are based on Natural Language Processing (NLP) techniques and rely on Machine Learning (ML), as studied by [16], or Deep Learning (DL) algorithms, as presented by [17], to identify entities and relationships from text. These ML and DL methods have demonstrated remarkable performance, even when applied to diverse domains such as biomedicine and cybersecurity [18,19]. However, they often require extensive human supervision and large training datasets as explained by [20,21]. In certain applications, supervised ontology population approaches are constrained by the availability of annotated training data. To address this limitation and reduce supervision effort, researchers have introduced unsupervised [18], semi-automatic [22], and hybrid approaches [23]. Several ontology population frameworks have been developed. For instance, ArtEquAKT is a web-based approach developed by [24] for populating ontologies in the domain of artists. OntoLearn [25] is a framework that extracts domain ontologies from web documents shared among virtual organization members. Other approaches include verb-based extraction for ontology population as demonstrated by [26].

2.2. Ontology population from an LLM perspective

Large Language Models (LLMs) offer new opportunities for creating semantic representations [10,27] or generating triples in RDF (Resource Description Framework) format [28]. These models operate based on requests or prompts, where prompt engineering involves designing specialized instructions to guide LLMs in generating output according to specified requirements [29]. Several LLMs have been introduced [30], including OpenAI’s GPT [31], Anthropic’s Claude [32], and Google’s Gemini [33], each trained on different datasets and developed for specific tasks. Comparative studies have evaluated these models’ effectiveness across different applications. [34] demonstrated that GPT excels in language translation, product descriptions, and transcript summaries, while Bard shows greater efficiency in information extraction and code-related tasks. [35] investigated Claude, GPT, Bard, and LLaMA across various applications, finding comparable effectiveness in named entity recognition and language comprehension tasks. Recent studies have analyzed LLMs’ capabilities in ontology population. [27] evaluated ChatGPT’s performance in extracting knowledge graphs from texts, while [10] proposed a benchmark for evaluating language models’ ability to generate ontology-guided knowledge graphs from natural language text. Additionally, [36] evaluated nine LLMs across three ontology learning tasks: term typing, taxonomy discovery, and the extraction of non-taxonomic relations.

3. Automatic ontology population from text

3.1. Case study

Fig. 1 illustrates our ontology population problem statement. Our case study focuses on the task of populating a domain ontology provided as input from French textual descriptions, each presenting an object from a given domain. We are particularly interested in such textual data because the sentences are short and incomplete, exhibiting some peculiar characteristics: sentences containing few or no verbs, object properties, data properties, few named entities, and n-ary relationships.²

² Complex relationships involving three or more classes, such as expressing that a real estate property is located at a certain distance from a given location.

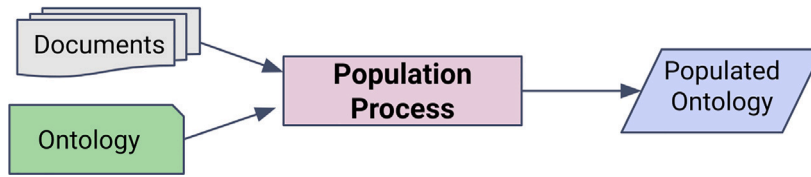


Fig. 1. Ontology population process.

We previously developed KOnPoTe [3] which is an ontology population approach that has been designed for such context. We explain briefly this approach in Section 3.3.

Although KOnPoTe has good performance in populating our domain ontology, its validation task remains time and human-resource consuming (more details in Section 4.2). Given the related work, the LLMs have the potential to contribute to automatically populate a domain ontology. However, it is necessary to investigate the extent to which LLMs are able to populate an ontology from short incomplete text. Since we give the full ontology as input to the LLM, we consider only LLMs that have large token size limitation. Among the state of art models that have shown interesting accuracy: Claude has the advantage of attaching a file of up to 50 MB to the prompt which is useful in our context. For the GPT, Gemini and LLaMa models, the size of our input surpasses the token size limitation.

3.2. Background and data

Our objective is to populate a domain ontology given as input with information contained in textual documents, where each document in the corpus is a textual description of an object in the domain under consideration. In our case, we consider three application domains: Real estate, Boats and Restaurants.

A domain ontology designates a formal, structured representation of knowledge that is designed by classes and their relationships. It allows us to specify classes, properties (attributes or relationships between classes), instances (concrete examples of classes) and axioms (logical rules). The latter define the semantics of the domain concerned. More formally, the ontology can be defined in our context as a tuple (C, P, I, A, R) where C is a set of classes, P a set of properties (object properties and data properties) characterizing the classes, I a set of individuals and assertions, A a set of axioms formulated in OWL (Web Ontology Language) and R a set of SWRL (Semantic Web Rule Language) rules. Ontology population refers to the process of adding elements to I .

We study three application domains:

The first one is real estate sales. The corresponding corpus is made up of house ads and the ontology describes knowledge about house sales. It contains classes such as the main class `Property` (designating the real estate), `House`, `Room`, etc. It also contains properties, for example the object property `contains` (linking a real estate property to its parts, or a house to its rooms), or the data property `areaInM2` (associating a house to a numerical value). The ontology also contains axioms, such as the fact that a `Property` can only be located in a maximum of one `Municipality`. Finally, SWRL rules are defined.

The second domain that we examine is boat sales. The corresponding corpus is made up of boat sales ads, and the ontology contains classes such as the main class `Boat` and other parts of the boat such as the class `Engine`, etc. It also contains properties, for example the object property `hasEquipment` (linking a boat to its equipments), or the data property `enginePower` (associating an engine to a numerical value).

The third domain is Restaurants. The corresponding corpus is made up of restaurants' descriptions, and the ontology represents knowledge about the restaurant domain. It contains classes such as the main class `Restaurant`, `Menu`, etc. It also contains properties, for example the object property `hasPrice` (linking a menu or a dish to its price), or the data property `nbPlaces`, associating a `Restaurant` to the maximum number of places.

Table 1 presents a comparison between the ontologies. We notice that the real estate and the restaurant ontology are more sophisticated than the boat ontology. Indeed, it contains more classes, data properties and object properties. Moreover, the real estate ontology has a higher number of properties domain and range class expressions. A class expression is an expression involving several classes. For example, the domain of the object property `contains` is represented by the expression of the union of the classes `Property` and `House`. Additionally, the boat ontology has a hierarchy with smaller depth (2 sub-class levels and 1 sub-property level). One last distinguished difference is that the boat ontology does not contain neither n-ary relationships nor SWRL rules.

Regarding the textual descriptions, all corpora are composed of 20 French classified ads per domain that were extracted from websites³.

Table 2 shows the ontologies and the corpus sizes. We notice that the restaurant ontology is the largest compared to the other ontologies, due to the number of classes and properties represented in this ontology, as presented in Table 1. The size of the text is relatively medium, with the maximum number of characters being 25,000 for the boat corpus, as shown in Table 2. The number of documents remains limited (see the explanation of the time-consuming evaluation task in Section 4.2), but it is sufficient to

³ The house ads were extracted from: <https://www.lecoindelimmo.com/> the boat ads from: <https://www.hisse-et-oh.com/>, and the restaurant ads from: <https://www.normandie-tourisme.fr/>

Table 1
Comparison between the Real estate, Boat and Restaurant ontologies.

Comparison criteria	Real Estate	Boats	Restaurants
Nb Classes	65	40	116
Nb Object Properties	14	11	17
Nb Data Properties	31	23	21
Nb Domain Class Expressions	7	0	1
Nb Range Class Expressions	11	11	6
Class hierarchy Depth	4	2	3
Property hierarchy Depth	2	1	1
N-ary relationships	Yes	No	Yes
SWRL rules	Yes	No	Yes

Table 2
Comparison between input size for the three domains.

Data	Token number			Character number		
	Ontology	Corpus	Input	Ontology	Corpus	Input
Real Estate	28K	6,673	34K	108K	20K	128K
Boats	37K	8,976	46K	105k	25K	130K
Restaurant	51K	6,249	57K	124K	19K	143K

establish a complete comparison between both categories of ontology population approaches. These different documents provide the necessary examples to demonstrate the strengths and limitations of both approaches, enabling a meaningful comparison of their differences.

We present hereafter the approaches tested: KOnPoTe and the LLM-based approach.

3.3. KOnPoTe

We developed KOnPoTe, a semantic approach that automatically populates a domain ontology from textual descriptions of an object. It stands for "Knowledge graph/ON-tology POpulation from TExts". Fig. 2 summarizes the main steps of this approach. In this section, we give a brief explanation of KOnPoTe. More details are described in [3]. Firstly, a terminology comparator is applied to both the text and the ontology. It aims to create correspondences between text mentions and ontological entities (classes, properties, and individuals), taking into account the syntactic variations in keywords. For example, it detects the classes House, Garden, etc. in the document shown in Fig. 2. Next, two analysis steps are implemented: a text-based analysis and a knowledge-based analysis.

The text-based analysis examines the correspondences obtained and uses textual indicators to populate the ontology. First, it populates the ontology according to the types of ontological entities detected in the correspondences. For example, a match with a class instantiates that class, while a match with a property instantiates that property, using the text to find the most appropriate subject and object of assertions. Further treatments are carried out, for example, to associate individuals from successive matches within the same sentence. Only the assertions that do not generate inconsistencies are added to the ontology. In the example shown in Fig. 2, this step creates, among other things, instances of the classes House, Garden, LivingRoom, and an assertion of the data property areaInM2 for this room with the value 50.

The knowledge-based analysis does not exploit any textual indicators but relies on the ontology knowledge and the set of assertions already created in the text-based analysis. It aims to ensure that the graph representing object property assertions of one document is as close as possible to the shape of a star, where the center corresponds to the individual representing the subject of the document in question (for example, the real estate property described in the document, in other words Property1 in the example from Fig. 2). In this step also, only the consistent properties are added to the ontology. In the example shown in Fig. 2, this step adds, among other things, the assertions of the object property contains that link the instance of the class House to the various instances of the Room sub-classes (BedRoom, LivingRoom, etc.).

3.4. LLM-based approach

As the research area of LLMs is evolving rapidly in many ways and applications, several models and techniques have been published [37–39]. Among the benchmark of released LLMs, we have chosen to launch experiments on multilingual models that have shown remarkable zero-shot abilities in natural language processing tasks and that have large token sizes since we are dealing with French textual data and giving an ontology as input.

The model Claude is the most suitable regarding the following case study constraints:

- Language: Claude is a multilingual model and can handle French data.
- Input nature: Compared to the rest of the models, Claude has a high window context and token size limitation that covers our input size (refer to Table 2).

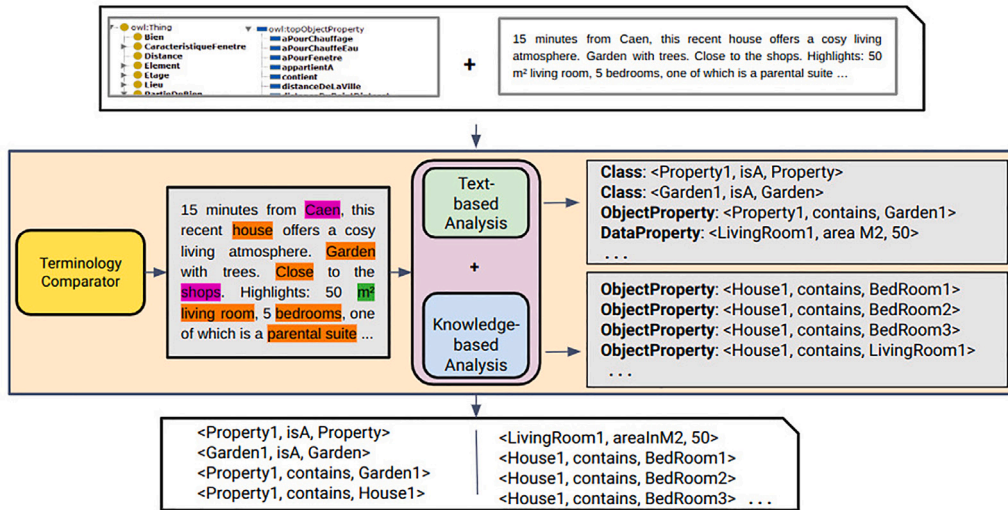


Fig. 2. Outline of the KOnPoTe approach, applied to an example document.

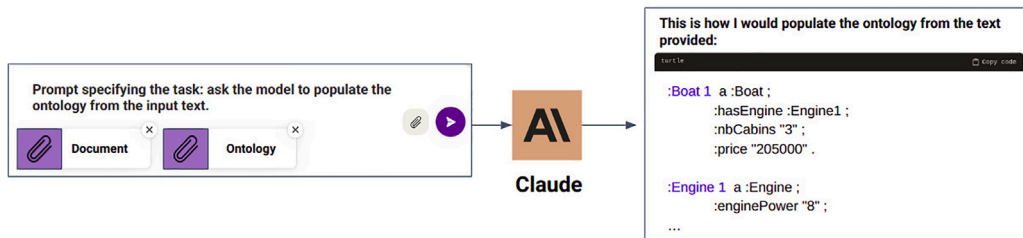


Fig. 3. Outline of the LLM-based approach, applied to an example document.

Our approach, illustrated in Fig. 3, uses the models in a zero-shot pipeline, which means that we exploit the model’s pre-acquired linguistic understanding, obtained during its general training, without providing it with specific training for the population task. We designed the approach to take into account the following elements, which specify our objective:

- Input: ontology in Turtle format + the textual descriptions.
- Output: populated ontology (Turtle format).
- Task: automatic ontology population from text.
- Language: French is the language used in the prompt, the ontology entities, and the text.

The LLM performs according to the input prompt. We would like to point out that each model has a specific prompting technique, and models may process the same task differently using the same prompt. Therefore, a prompt engineering step, detailed in Section 4.1, is established to determine the most efficient prompt for our task. This step is deemed indispensable in the case of such an approach, as the model’s response is highly dependent on the input prompt.

The experimental protocol, detailed in Section 4.2, presents the different prompts used. Prompt engineering involves the use of a user-guided schema during the prompting process, in order to improve the consistency of the triples generated and integrate domain-specific knowledge to enhance the fidelity of the triples generated by the LLM.

We have tested the version Claude-Instinct-1.2 of this model and interrogated it directly using a zero-shot prompt, in other words, without any prior prompt tuning (few-shot, chain-of-thought prompting, etc.) and without any model fine-tuning, in order to ensure a fair comparison with KOnPoTe. The approach takes as input a prompt containing a textual explanation of the task to be performed, as well as a corpus document and the ontology.

Note that this approach is equivalent to KOnPoTe in terms of input, output, and treatments (we consider the population process as a black-box). The LLM generates triples according to the input prompt. A prompt engineering step, detailed in Section 4.1, is established to determine the most efficient prompt for our task.

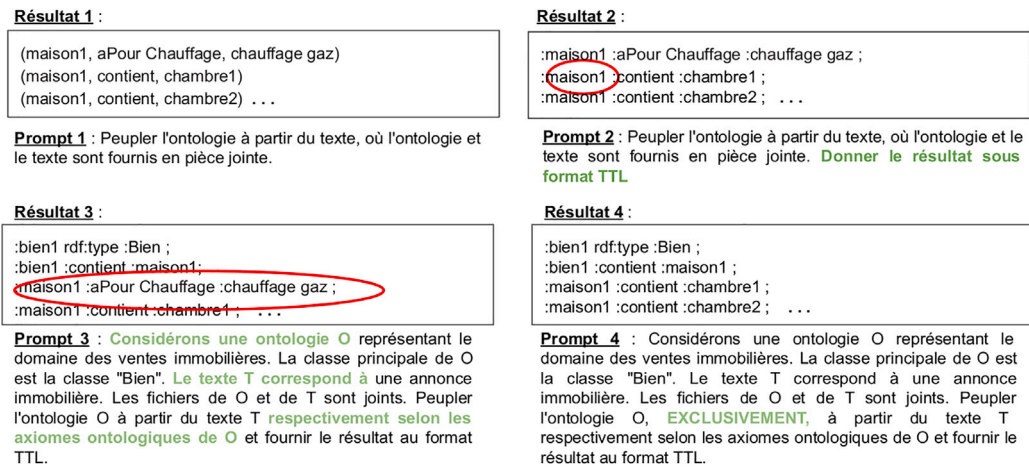


Fig. 4. Original French prompts applied to the real estate domain.

4. Experiments and comparison

This section presents our experiments on the domains of real estate, boat sales, and restaurants. KOnPoTe is tested via the executable provided in [3]. The Claude-Instant version is tested directly via the API.⁴

4.1. Prompt engineering

The response generated by an LLM is highly dependent on the prompt given to the model. In this section, we detail the prompt engineering used to determine the most appropriate prompt. Several works have discussed the impact of prompt engineering on LLM performance and the variance of responses generated for the same task using different prompts. [40,41] have shown that the response has a strong correlation with the prompt given as input. [42] have proposed a catalog of prompt engineering techniques, so-called patterns, which have been applied when interacting with ChatGPT. These techniques are proposed and tested only on the LLMs GPT and Bard. However, these patterns have not, to our knowledge, been tested on Claude, which is addressed in this paper.

In addition to that, the prompt engineering techniques described were developed primarily for English-language prompts, and their effectiveness has not been evaluated for prompts in other languages. In our study, we aimed to interrogate the French-language capabilities of the Claude model, as we were working with French-language data (ontologies and text). Our preliminary experiments showed that when prompting the model using zero-shot English prompts, the generated results were translated to English. Specifically, the IRIs (Internationalized Resource Identifiers) that were initially in French were translated to English by Claude. Therefore, we decided to conduct our experiments by prompting the model in French, in order to generate results directly in that language.

We illustrate in Fig. 4 the original French prompts used in our experiments and provide in Fig. 5 their English translation. The use of French-language prompts allowed us to obtain French-language outputs from the model, without requiring an additional translation step. This approach was crucial for our study, as we aimed to evaluate the model's performance on the original French-language data, rather than on translated content.

We create the prompt progressively by studying the response given by each prompt specifying how to populate the ontology domain. Fig. 5 illustrates the 4 prompts used (translated to English) and the result generated by Claude for each prompt to populate the ontology from the document presented in Fig. 2.

Prompt 1 is a basic prompt that specifies the task only. It generates triples of the form (subject, predicate, object) from ontology instances. However, the format of the result of this prompt cannot be directly exploited in the validation step. To remedy this, we specify the desired format, Turtle, in Prompt 2.

Analyzing the result of Prompt 2, we notice that the LLM did not respect the ontological axioms given in the input ontology, did not automatically detect the main class `Property` of the ontology, and did not generate the assertions related to this class (its parts, price, etc.). At this stage, the LLM is unable to detect this class, as it is not aware of the context. According to [42], the model lacks the context of the words used in the prompt. We have clarified this context in Prompt 3 and explicitly instructed the LLM to respect the ontological axioms.

At this stage, Prompt 3 populates the classes in our ontology, but it also generates information that is not contained in the text. For example, the result of Prompt 3 in Fig. 5 mentions that the property has gas heating and an electric water heater, but this

⁴ <https://www.anthropic.com/api>

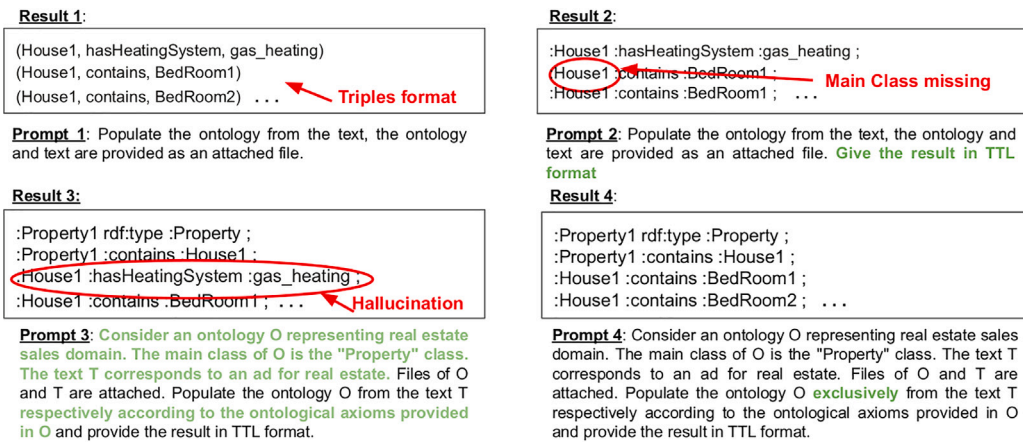


Fig. 5. English translation of the prompt engineering steps.

information is not mentioned in the ad text. The addition of these extra instances can be due to an LLM hallucination. To limit this confusion, we explicitly specified in Prompt 4 to populate the ontology only with information mentioned in the input text, by adding the word “exclusively” to Prompt 3, which reduced the number of hallucinated assertions. This last prompt (Prompt 4) has been tested on the real estate domain and has been adapted for the other domains. Prompt 4 remains the most suitable for our automatic ontology population task. This prompt is used and evaluated according to the experimental protocol detailed hereafter.

4.2. Experimental protocol

The house sales corpus and the real estate ontology used in our tests are provided by KOnPoTe,⁵ except that we only considered the first 20 ad descriptions in this corpus. The results obtained for the boat sales and restaurant domains, as well as those of the Claude-based approach, are available online.⁶ They are evaluated using the same protocol as [3].

A Gold Standard (GS) is established for each domain to evaluate the accuracy of both techniques. This GS serves as a reference ontology with validated labels representing the textual corpus for each domain. We manually reviewed the textual corpus to populate the GS.

We compared the assertions generated by both approaches with those of the GS for both domains. We calculate, for each domain, precision (Prec.), recall (Rec.), and F1-score (F1-sc.) using the following formulas:

$$Prec. = \frac{TP}{TP + FP} \quad Rec. = \frac{TP}{TP + FN} \quad F1 - sc. = \frac{2 \times Prec. \times Rec.}{Prec. + Rec.}$$

A true positive (TP) is a correct assertion, a false positive (FP) is an erroneous assertion, and a false negative (FN) is a missing assertion. Both macroscopic (mac) and microscopic (mic) values are calculated for each evaluation metric. The macroscopic score averages the metrics for each ad (all ads have the same weight), while the microscopic score takes into account the sum of all TP, FP, FN (all assertions have the same weight). Thus, an ad generating many assertions has more impact on the microscopic metric, while an ad generating few assertions has more impact on the macroscopic calculation. We also check the consistency of the resulting ontologies using the Pellet reasoner.⁷

We decided to launch experiments on only 20 ads per domain, since the evaluation of both approaches is time- and resource-consuming. Indeed, the creation of the GS for each domain is not the only labor-intensive task; there is also the comparison between each ad’s assertion and the corresponding GS assertion to determine the correct, mistaken, and missing ones. This comparison task involves three verification rules:

1. We ignore the class instance assertions since they are often redundant with property assertions. For example, with the assertion (`Distance1`, `distanceToTown`, `Caen`), we know via the definition of the domain of `distanceToTown` that `Distance1` is an instance of the class `Distance`.
2. We ignore properties that lead to counting the same element multiple times. For example, if there is a property and its inverse, the reasoner translates an assertion of the first property into an inverse assertion of the other. We end up with two equivalent assertions. In this case, all assertions of one of the two properties are ignored. As another example, we can cite the case where a property is never populated by itself but only through its sub-properties. In this case, we ignore this property so that an assertion (true or false) does not count twice, as inferred assertions are taken into account.

⁵ <https://zenodo.org/records/5776752>

⁶ <https://zenodo.org/records/11664135>

⁷ <https://github.com/stardog-union/pellet>

Table 3
Results of the two approaches on both domains.

Domain	Approach	Prec. (mac)	Rec. (mac)	F1-sc. (mac)	Prec. (mic)	Rec. (mic)	F1-sc. (mic)
Real Estate	KOnPoTe	0.94	0.90	0.92	0.93	0.87	0.90
	Claude	0.77	0.59	0.66	0.77	0.56	0.65
Boats	KOnPoTe	0.91	0.92	0.92	0.91	0.92	0.92
	Claude	0.77	0.75	0.76	0.80	0.75	0.77
Restaurants	KOnPoTe	0.92	0.77	0.82	0.90	0.71	0.80
	Claude	0.73	0.62	0.65	0.82	0.67	0.75

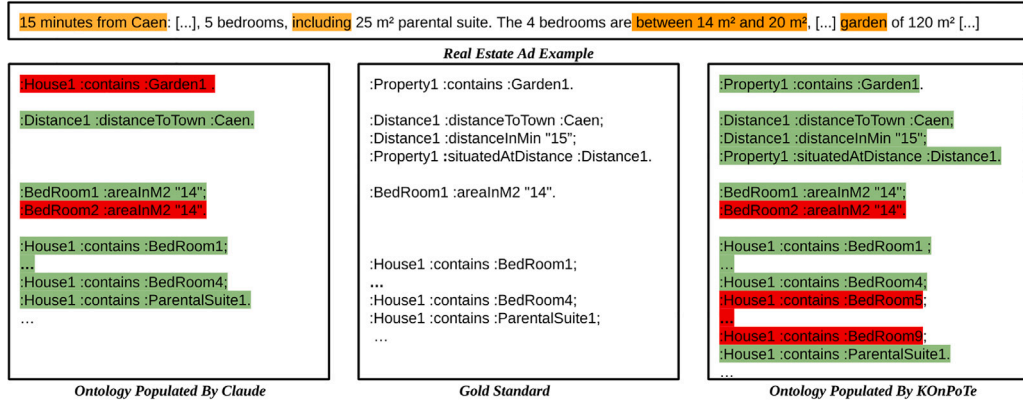


Fig. 6. Example of real estate ontology populated by Claude and KOnPoTe from a house sale ad and the corresponding Gold standard.

- The approaches may generate equivalent individuals without detecting that they are equivalent. In this case, we assume that an equivalence axiom (owl:sameAs) is missing and we count it as a missing assertion (an FN). This avoids counting twice an erroneous FP and FN.

To summarize, this evaluation step involves determining, for each instance in each description text from each domain generated by both approaches, prior equivalences with the GS instances. We also need to calculate the number of missing sameAs links in case of undetected equivalences and to ignore assertions that repeatedly represent the same property. More explanation about this evaluation step is presented in [3].

4.3. Results and discussion

Table 3 shows the results of both approaches. The KOnPoTe approach outperforms the Claude model-based approach on all domains. Indeed, KOnPoTe achieves better results on all the metrics calculated. The Claude-based approach has a relatively lower precision (73%–82%), which means that some of the assertions generated are false (around 20%). Moreover, recall is low (59%–75%), indicating that many assertions are missing (almost 25%).

Fig. 6, Fig. 7, and Fig. 8 show examples of real estate, boat, and restaurant texts, the assertions generated by each approach, and the corresponding GS for each example.

When analyzing the examples, we notice that Claude has generated some mistaken assertions (object properties with wrong range and domain) compared to the GS and KOnPoTe. Indeed, the real estate ontology provided as input specifies that only instances of Property can contain instances of PropertyPart. The class Garage is a subclass of PropertyPart, so Garage1 described in Fig. 6 can only be contained by a Property. However, we can see that the individual Garage1 is contained by an instance of House, and not an instance of Property, which makes the ontology inconsistent according to the Pellet reasoner. On the contrary, KOnPoTe has created a consistent and correct assertion.

An example of an n-ary relationship is illustrated in Fig. 6. The example shows that the real estate is located 15 min from Caen. We can see that the population is correctly done by KOnPoTe: Property1 is located at a distance distance1, and distance1 is associated with the notion of 15 minutes and with the city of Caen. For Claude, the distance1 instance created does indeed concern the city of Caen, but it is neither associated with Property1 nor with the notion of 15 minutes.

Another example of an n-ary relationship is illustrated in Fig. 8. The example indicates that the restaurant opens every day at 11 am and closes at 2 pm. Neither KOnPoTe nor Claude populated it correctly. In fact, both approaches detected only the OpenTime1 or OpenHour1 but did not instantiate the rest of the properties (startHour and endHour). The correct n-ary relationship, as instantiated in the GS, is that Restaurant1 has an opening time OpenTime1, and this latter has an opening hour OpenHour1 which starts at startHour 11 and ends at endHour 2.

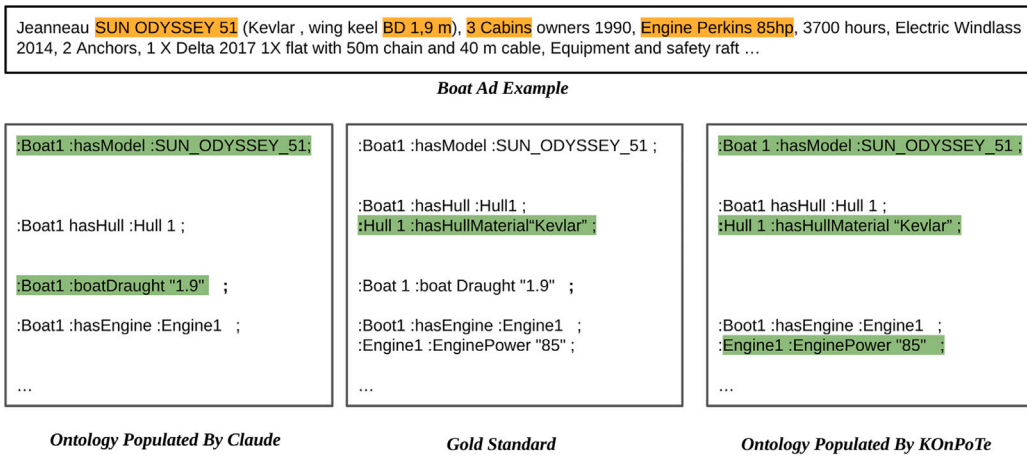


Fig. 7. Example of boat ontology populated by Claude and KOnPoTe from a boat sale ad and the corresponding Gold standard.

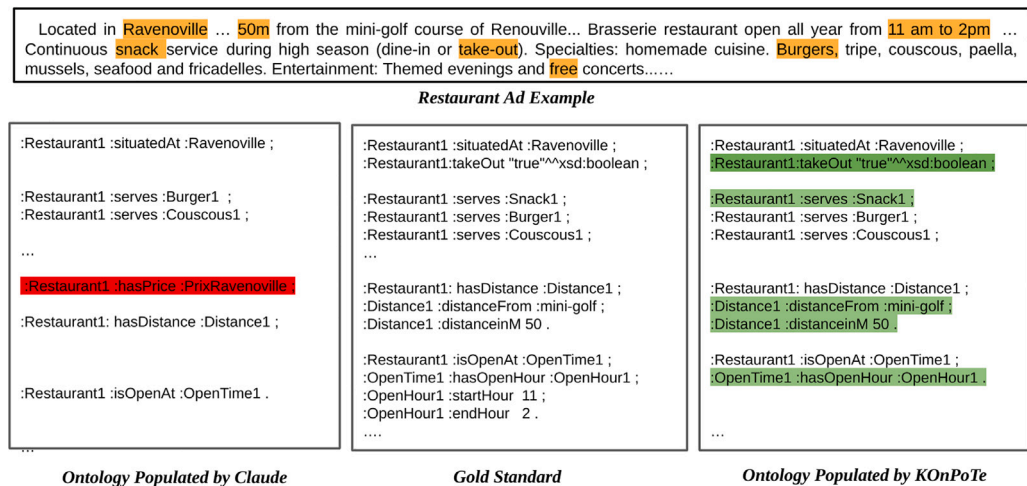


Fig. 8. Example of Restaurant ontology populated by Claude and KOnPoTe from a restaurant description and the corresponding Gold standard.

On the other hand, KOnPoTe tends to generate redundant individuals. In the example shown in Fig. 6, it generates 9 bedroom instances and one parental suite instance, whereas Claude generates, rightly, only 4 bedroom instances and one parental suite instance. However, Claude does not tend to instantiate all the classes and individuals. In fact, when analyzing the missing assertions, we find that most of them are missing instances of classes or data properties. An example of a missing class instance assertion is illustrated in Fig. 8. *Restaurant1* serves a lot of dishes, and some of them are special or signature dishes. The model instantiates only the most common dishes but not the lesser-known ones (e.g., *Snack1* in Fig. 8 which is special vs. *Burger1* that is a more common dish). The same problem is detected when instantiating the boat equipment: for example, a boat that has a lot of sophisticated equipment, the model recognizes and populates only the common ones (radar, GPS, etc.). On the other hand, KOnPoTe instantiates all the classes and individuals if they are present in the input ontology.

Another example of a missing property instance assertion is illustrated in Fig. 7, where the data property *enginePower* with a value of 85 is not instantiated by Claude, in contrast to KOnPoTe.

Nevertheless, Claude is more robust to syntax variations and performs better in detecting abbreviations. Indeed, in the example shown in Fig. 7, it detects *BD* as Boat Draught and instantiates the data property *boatDraught* with the correct value, while KOnPoTe does not detect it.

A detailed analysis of all the generated domain ontologies has enabled us to deduce the strengths and limitations of both approaches. These are summarized in Table 4.

KOnPoTe performs rather well but often generates several equivalent individuals. Conversely, Claude does not generate this type of redundancy but, instead, produces inconsistent assertions in terms of reasoning. Indeed, the ontology generated by Claude is inconsistent for 90% of the real estate ads, 60% of the boat ads, and 75% of the restaurant ads. This inconsistency arises because

Table 4
Comparison of the strength and limitations of Claude and KOnPoTe.

Approach	Highlights	Limitations
KOnPoTe	-Consistency of results -Explainability of results -Detection of n-ary relationships	-Redundancy of generated individuals -Confusion in complicated cases
Claude	-No redundancy of individuals -Detection of abbreviations	-Inconsistency of results -Inexplainability of results -Confusion in complicated cases

the domain and range axioms are not always respected by Claude. On the other hand, such errors are not possible with KOnPoTe, as each step of the approach ensures that candidate assertions do not generate inconsistencies, and they are excluded if they do.

When it comes to assertions of n-ary properties, neither approach performs perfectly, but KOnPoTe handles them relatively well, whereas Claude struggles more, particularly in linking individuals together. However, the LLM-based Claude has the advantage of detecting abbreviations in the text, while KOnPoTe is not capable of detecting them unless they are explicitly mentioned in the input ontology.

Additionally, both approaches fail in the most complicated cases, which require a deep understanding of the text, as such information may not be explicitly expressed or easily detected from the text itself.

5. Conclusions and future work

This work presents a comparative study of two different approaches for automatic ontology population: a semantic approach using KOnPoTe and an approach based on large language models (LLMs) using Claude. The study demonstrates that each approach has its own strengths and limitations.

The semantic KOnPoTe approach leverages the reasoning capabilities of ontologies, making it more efficient at maintaining consistency and logical coherence within the populated ontology. However, it has certain limitations, notably a tendency to introduce redundant individual assertions. On the other hand, the LLM-based approach excels at detecting implicit information but is less adept at handling consistency constraints and does not generate redundant individuals.

Additionally, KOnPoTe has been specifically developed for the context of French descriptive texts, which was the use case evaluated in this study. In other contexts involving different types of texts or languages, an alternative approach may need to be implemented. A key advantage of the LLM approach is the relative simplicity of its application via Claude's available API, making it more immediately deployable compared to the design and implementation effort required for a semantic approach customized to the characteristics of the document corpus.

An interesting perspective for future work is the development of a hybrid approach that combines the strengths of both methods. Such an approach could leverage the LLM to propose candidate assertions, which would then be vetted and potentially corrected by enforcing consistency using ontological knowledge and the reasoning capabilities of KOnPoTe. This would allow the LLM to contribute its generation capabilities while ensuring logical coherence. The hybrid approach could further incorporate a knowledge-based analysis module, inspired by KOnPoTe, to supplement the ontology with consistent assertions that may be missed by the LLM.

Ultimately, this comparative analysis highlights the complementary nature of semantic reasoning approaches and more flexible, language model-driven methods for the challenging task of ontology population from text. A synergistic combination of these paradigms presents an attractive path toward more robust and context-adaptable ontology population systems.

CRedit authorship contribution statement

Aya Sahbi: Writing – original draft, Methodology, Conceptualization. **Céline Alec:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Conceptualization. **Pierre Beust:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

The original French version of Fig. 2, Fig. 6, Fig. 7 and Fig. 8 is illustrated respectively by Fig. 9, Fig. 10, Fig. 11, and Fig. 12.

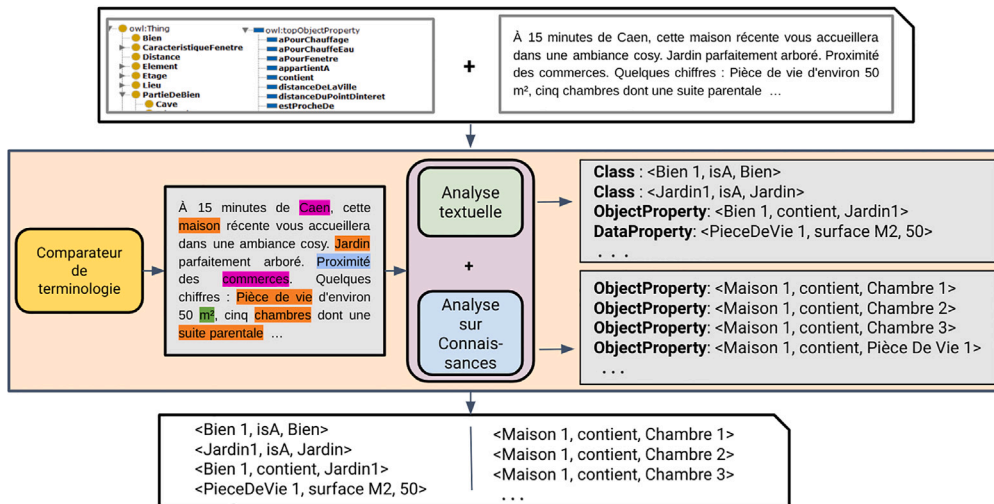


Fig. 9. French version of Fig. 2 (KOnPoTe outline).

À 15 minutes de Caen, cette maison récente vous accueillera dans une ambiance cosy. Jardin parfaitement arboré. Proximité des commerces. Quelques chiffres : Pièce de vie d'environ 50 m², cinq chambres dont une suite parentale au rez de chaussée de 25 m². Les quatre chambres sont de 14 m² à 20 m² avec dressing, deux salles de bains, une salle de douche, garage de 35 m², le tout sur un terrain clos et arboré de 1017 m² Honoraires : 4 % TTC charge acquéreur 595 000 €.

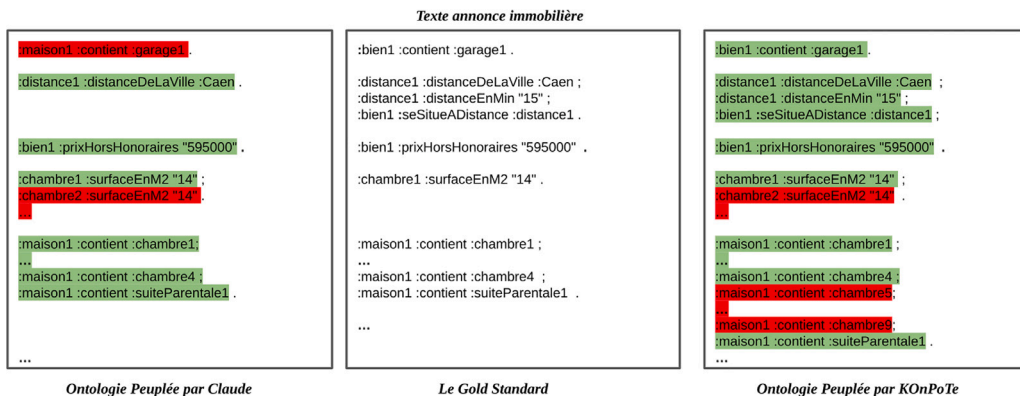


Fig. 10. French version of Fig. 6 (real estate ontology population example).

Jeanneau SUN ODYSSEY 51 (Kevlar, quille ailée TE 1,9 m), 3 Cabines propriétaires 1990, Moteur Perkins 85 cv, 3700 heures, Guindeau électrique 2014, 2 Ancres, 1 X Delta 2017 1X plate avec 50m de chaîne et 40 m de câble, Équipement et radeau de sécurité

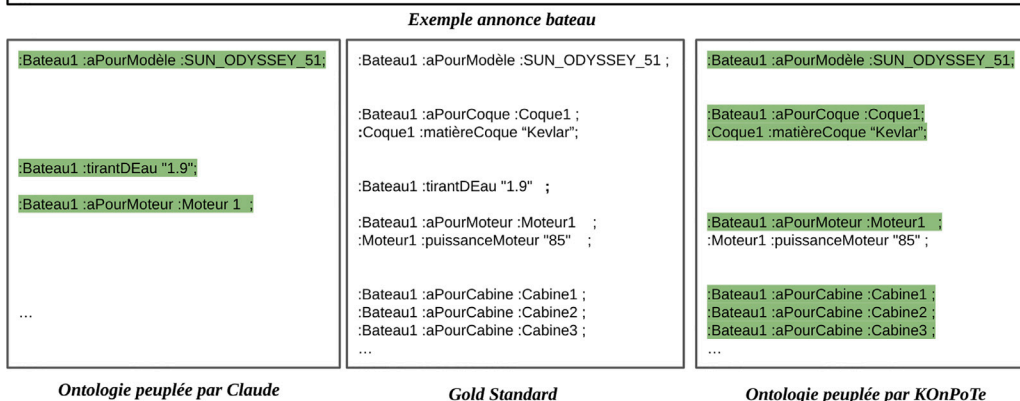


Fig. 11. French version of Fig. 7 (boat ontology population example).

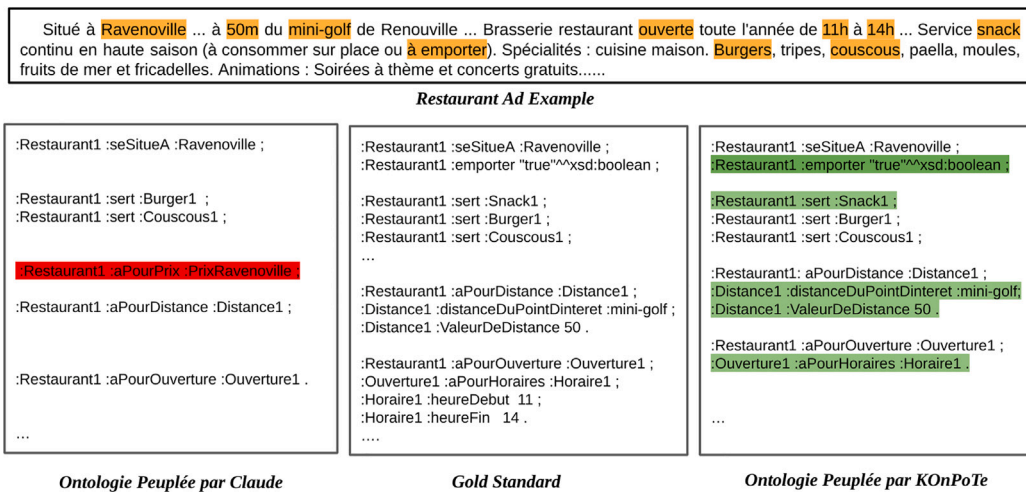


Fig. 12. French version of Fig. 8 (restaurant ontology population example).

Data availability

Data will be made available on request.

References

- [1] A.N.E. Sahbi, C. Alec, P. Beust, Peuplement automatique d'ontologie: l'IA générative est-elle plus efficace qu'une approche sémantique?, in: 24ème conférence francophone sur l'Extraction et la Gestion des Connaissances (EGC), 2024.
- [2] A. Sahbi, C. Alec, P. Beust, Automatic ontology population from textual advertisements: Ilm vs. semantic approach, *Procedia Comput. Sci.* (ISSN: 1877-0509) 246 (2024) 3083–3092, <http://dx.doi.org/10.1016/j.procs.2024.09.364>, <https://www.sciencedirect.com/science/article/pii/S1877050924023925>, 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024).
- [3] C. Alec, Ontology population from french classified ads, in: *International Conference on Conceptual Structures*, Springer, 2023, pp. 155–170.
- [4] Z. Zhang, F. Ciravegna, Named entity recognition for ontology population using background knowledge from wikipedia, in: *Ontology Learning and Knowledge Discovery using the Web: Challenges and Recent Advances*, IGI Global, 2011, pp. 79–104.
- [5] I. Muhammad, A. Kearney, C. Gamble, F. Coenen, P. Williamson, Open information extraction for knowledge graph construction, in: *Database and Expert Systems Applications: DEXA 2020 International Workshops BIODDD, IWCFs and MLKgraphs*, Bratislava, Slovakia, September 14–17, 2020, *Proceedings 31*, Springer, 2020.
- [6] V.D. Boer, M. Someren, B. Wielinga, Relation instantiation for ontology population using the web, 2006, pp. 202–213, http://dx.doi.org/10.1007/978-3-540-69912-5_16.
- [7] S. Sahnoun, S. Elloumi, S. Ben Yahia, Event detection based on open information extraction and ontology, *J. Inf. Telecommun.* 4 (3) (2020) 383–403.
- [8] K. Adnan, R. Akbar, An analytical study of information extraction from unstructured and multidimensional big data, *J. Big Data* 6 (1) (2019) 1–38.
- [9] J.L. Martinez-Rodriguez, A. Hogan, I. Lopez-Arevalo, Information extraction meets the semantic web: a survey, *Semant. Web* 11 (2) (2020) 255–335.
- [10] N. Mihindukulasooriya, S. Tiwari, C.F. Enguix, K. Lata, Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text, in: *International Semantic Web Conference*, Springer, 2023, pp. 247–265.
- [11] C. Biemann, Ontology learning from text: A survey of methods, *J. Lang. Technol. Comput. Linguist.* 20 (2) (2005) 75–93.
- [12] P. Buitelaar, P. Cimiano, B. Magnini, *Ontology Learning from Text: Methods, Evaluation and Applications*, vol. 123, IOS Press, 2005.
- [13] M. Lubani, S.A.M. Noah, R. Mahmud, Ontology population: Approaches and design aspects, *J. Inf. Sci.* 45 (4) (2019) 502–515.
- [14] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, E. Zavitsanos, Ontology population and enrichment: State of the art, in: *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution: Bridging the Semantic Gap*, Springer, 2011, pp. 134–166.
- [15] W. Wong, W. Liu, M. Bennamoun, Ontology learning from text: A look back and into the future, *ACM Comput. Surv. (CSUR)* 44 (4) (2012) 1–36.
- [16] A. Imsonbut, C. Sirikayon, An alternative technique for populating thai tourism ontology from texts based on machine learning, in: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science, ICIS, IEEE*, 2016, pp. 1–4.
- [17] P. Sambandam, D. Yuvaraj, P. Padmakumari, S. Swaminathan, Spiking equilibrium convolutional neural network for spatial urban ontology, *Neural Process. Lett.* (2023) 1–20.
- [18] A. Ayadi, A. Samet, F.d. de Beuvron, C. Zanni-Merk, Ontology population with deep learning-based NLP: a case study on the biomolecular network ontology, *Procedia Comput. Sci.* 159 (2019) 572–581.
- [19] H. Gasmı, J. Laval, A. Bouras, Cold-start cybersecurity ontology population using information extraction with LSTM, in: *2019 International Conference on Cyber Security for Emerging Technologies, CSET, IEEE*, 2019, pp. 1–6.
- [20] V. Jayawardana, D. Lakmal, N. de Silva, A.S. Perera, K. Sugathadasa, B. Ayesha, M. Perera, Semi-supervised instance population of an ontology using word vector embedding, in: *Int. Conference on Advances in ICT for Emerging Regions, ICTer 17, IEEE*, 2017, pp. 1–7.
- [21] F. Suchanek, G. Ifrim, G. Weikum, LEILA: Learning to extract information by linguistic analysis, in: *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, 2006, pp. 18–25.
- [22] A. Maedche, S. Staab, Ontology learning for the semantic web, *IEEE Intell. Syst.* 16 (2) (2001) 72–79.
- [23] S. Castano, I.S.E. Peraldi, A. Ferrara, V. Karkaletsis, A. Kaya, R. Möller, S. Montanelli, G. Petasis, M. Wessel, Multimedia interpretation for dynamic ontology evolution, *J. Logic Comput.* 19 (5) (2009) 859–897.

- [24] H. Alani, S. Kim, D.E. Millard, M.J. Weal, W. Hall, P.H. Lewis, N.R. Shadbolt, Automatic ontology-based knowledge extraction from web documents, *IEEE Intell. Syst.* 18 (1) (2003) 14–21.
- [25] R. Navigli, P. Velardi, Learning domain ontologies from document warehouses and dedicated web sites, *Comput. Linguist.* 30 (2) (2004) 151–179.
- [26] J. Makki, A.-M. Alquier, V. Prince, Ontology population via NLP techniques in risk management, *Int. J. Humanit. Soc. Sci. (IJHSS)* 3 (3) (2009) 212–217.
- [27] M. Trajanoska, R. Stojanov, D. Trajanov, Enhancing knowledge graph construction using large language models, 2023, Preprint [arXiv:2305.04676](https://arxiv.org/abs/2305.04676).
- [28] R.R. Souza, T. Pinheiro, J. Oliveira, J.C. dos Reis, Knowledge graphs extracted from medical appointment transcriptions: Results generating triples relying on LLMs, 2023, pp. 129–139.
- [29] Y. Zhou, A.I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba, Large language models are human-level prompt engineers, 2022, arXiv preprint [arXiv:2211.01910](https://arxiv.org/abs/2211.01910).
- [30] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, J. rong Wen, A survey of large language models, 2023, [http://dx.doi.org/10.48550/arXiv.2303.18223](https://dx.doi.org/10.48550/arXiv.2303.18223), ArXiv abs/2303.18223.
- [31] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, 2023, arXiv preprint [arXiv:2303.08774](https://arxiv.org/abs/2303.08774).
- [32] A. Anthropic, Introducing the next generation of claude, 2024.
- [33] S. Pichai, D. Hassabis, Introducing gemini: our largest and most capable AI model, 2023.
- [34] I. Ahmed, M. Kajol, U. Hasan, P.P. Datta, A. Roy, M.R. Reza, ChatGPT vs. Bard: A Comparative Study, *UMBC Stud. Collect.* (2023).
- [35] A. Borji, M. Mohammadian, Battle of the Wordsmiths: Comparing ChatGPT, GPT-4, Claude, and Bard, 2023, GPT-4, Claude, and Bard (June 12, 2023).
- [36] 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.
- [37] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, 2024, arXiv preprint [arXiv:2402.06196](https://arxiv.org/abs/2402.06196).
- [38] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, X. Hu, Harnessing the power of llms in practice: A survey on chatgpt and beyond, *ACM Trans. Knowl. Discov. Data* (2023).
- [39] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, *ACM Trans. Intell. Syst. Technol.* (2023).
- [40] I. Arawjo, C. Swoopes, P. Vaithilingam, M. Wattenberg, E. Glassman, ChainForge: A visual toolkit for prompt engineering and LLM hypothesis testing, 2023, Preprint [arXiv:2309.09128](https://arxiv.org/abs/2309.09128).
- [41] A. Gao, Prompt engineering for large language models, 2023, Available at SSRN 4504303.
- [42] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D.C. Schmidt, A prompt pattern catalog to enhance prompt engineering with chatgpt, 2023, Preprint [arXiv:2302.11382](https://arxiv.org/abs/2302.11382).

Aya Sahbi is a Ph.D. student at the GREYC lab, University of Caen-Normandie (France). Her doctoral work focuses on the challenge of error detection and correction using automatic ontology population, which lies at the intersection of several cutting-edge fields including ontologies, knowledge graphs, and large language models.

Céline Alec is an associate professor in the CODAG group at the GREYC lab, University of Caen-Normandie (France). She has a Ph.D. in computer science from Paris-Saclay University (France). Her research interests include knowledge engineering, ontologies, the Semantic Web, semantic annotation of documents, and Linked Open Data.

Pierre Beust is a full professor of Computer Science at the University of Rennes. He is attached to the IRISA laboratory's SHADOC (Systems for Hybrid Analysis of DOCUMENTs) team for his research activities. He works on computerized environments for pedagogy, in particular on user activity traces and person-system interactions.