**EXPERIMENTS**

***PHASE 1***

In the first phase of the process of running the experiments, the LLMs were given three different command/prompts, with minor but important differences concerning the aim and purpose of the ontology. The first prompt is simple and short, simply asking the model to produce an ontology for the specific research field as well as to focus on 1-3 specific parameters that must be mentioned in a SAR ontology. The second prompt specifies the aim of the ontology, to help it build it, including again the same parameters that need to be addressed. The third and most comprehensive prompt is more precise and direct, asking the models to build an ontology, specifying its aim as well as its scope, considered to be right one that we will use in order to compare later with the hybrid model. All three prompts where given to the models in 2 different ways [[1]](#footnote-1). First, on a one-shoot prompt, and later on a chain-of-thoughts prompt, broken down to the 3 sentences that the prompt is consisted of.

**Prompt1:** *Build an ontology about Search and Rescue missions in wildfires incidents, you will reuse other existing ontologies. In the process you should focus on modeling different aspects of SAR missions, such as environmental conditions, search techniques and equipment, rescue operation teams, and victim identification. Give me the final result in ttl format.*

**Prompt2:** *Build an ontology about Search and Rescue missions in wildfire incidents. The aim of the ontology is to represent knowledge related to Search and Rescue missions (e.g., mission, IoT entities, sensors, first search and rescue teams, etc.). You will reuse other related ontologies, in the process, you should focus on modeling different aspects of SAR missions, such as environmental conditions, search techniques and equipment, rescue operation teams, and victim identification. Give the output in ttl format.*

**Prompt3:** *Build an ontology about Search and Rescue missions. The aim of the ontology is to represent knowledge related to Search and Rescue missions (e.g., mission, IoT entities, sensors, first responders, rescue teams, etc.) and the scope is wildfire in forests. You will reuse existing related ontologies, in the process, you should focus on modeling different aspects of SAR missions, such as environmental conditions, search techniques and equipment, rescue operation teams, and victim identification, data from heterogeneous sensors, etc. Give the output in ttl format.*

**Observations on the LLMs**

**GPT 3.5**

The most consistent and precise LLM as far as it concerns the given outcomes and results, but also on level of performance. Even though it’s free for users, it’s the most productive and efficient LLM compared to the other three, probably due to the many years of training and fine-tuning. Almost same metrics as the GPT-4, but during using 4 we faced some drawbacks that made us run again the prompts. It understood the command better on the one-shoot way rather than on chain-of-thoughts, never freeze or stopped the generating procedure, always gave the output file on the correct and asked format, perceived the concept of reusability along with GPT-4, and also gave the most individual examples from any other model.

**GPT 4**

On general base, GPT-4 had a worthy performance, but not on the same level as GPT 3.5. Half of the times that was asked to generate the ontology, especially on the third sentence of the chain-of-thought prompt, it stuck and stopped generating, so we had to run again the whole prompt from the start. On the second prompt, the chain-of-thoughts approach gave the same results as the one-shoot approach, perhaps because the two commands were executed successively. With GPT 3.5, it understood the concept of reusability on a adequate level. Except from one case on the second prompt, on every other experiment it gave lower metrics that GPT 3.5. Its response time was much bigger than GPT 3.5 but surely better that the other two models.

**Bard**

Bard had the best performance as far as it concerns axioms and classes, but the produced results weren’t always edible on Protégé. It gave faster results on the smaller and simpler sentences, but stopped many time the generating procedure on the half (almost at every experiment), so we had to run again the prompt, or ask to continue generating from the point it stopped. Compared to all the LLMs, it had the bigger time response. It didn’t really understand the concept of reusability, based on the given code, and it was the only model that didn’t produce any text first that explained the ontology and its parameters, it only produced the ttl format code, instead of the other three LLMs that all gave content on the one-shoot approach, as well as chain-of-thoughts. In order to include it in the evaluation process and acquire tangible results, we had to run the experiments at least 3 times at every approach (first two time the result wasn’t on an absolute level in turtle syntax but in a mixed type syntax).

**Llama 2**

The worst performing model. It seemed to understand the concept of the ontology that we needed to build, and at first it produced a layout of the ontology in natural language text, where all the possible parameters were noted, but it was unable to translate this knowledge to an ontology. The given code wasn’t representative to the introductive text, mostly poor, inadequate and incomplete, with one time, although it was asked to give the output file in ttl format, it produced a different format, so we had to ask it to translate it in turtle syntax. It didn’t understand the “ttl format” on the sentence, so we had to ask for turtle syntax. It had better results on more detailed sentences, whereas on a one-shoot approach it produced only natural language text and didn’t generate any code, even though the last sentence was this command.

**Evaluation Metrics on the LLM generated ontologies**

We kept on using the third prompt for our experiments as the one of the most precise and inclusive of the three (described and analyzed the aim and scope of the wanted ontology in a clear and definite way). The metrics on classes and object properties of the generated ontologies are pictured on the following tables, and mapped with the “Gold” Ontology[[2]](#footnote-2) that we chose to compare with.

**Table 1:** Evaluation metrics on Classes.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Classes** | **True Positives** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 80 |  |  |  |  |  |  |
| **Bard-One-shoot** | 19 | 8 | 11 | 72 | 42% | 10% | 0,16162 |
| **Bard-Chain-of-thoughts** | 8 | 4 | 4 | 76 | 50% | 5% | 0,09091 |
| **GPT-3,5-One-shoot** | 6 | 4 | 2 | 76 | 67% | 5% | 0,09302 |
| **GPT-3,5-Chain-of-thoughts** | 14 | 7 | 7 | 73 | 50% | 8,75% | 0,14894 |
| **GPT-4-One-shoot** | 8 | 6 | 2 | 74 | 75% | 7,5% | 0,13636 |
| **GPT-4-Chain-of-thoughts** | 5 | 4 | 1 | 76 | 80% | 5% | 0,09412 |
| **Llama2-One-shoot** | - | - | - | - | - | - | - |
| **Llama2-Chain-of-thoughts** | - | - | - | - | - | - | - |

**Table 2:** Evaluation metrics on object properties.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Obj. properties** | **True Positive** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 60 |  |  |  |  |  |  |
| **Bard-One-shoot** | 4 | 1 | 3 | 59 | 25% | 1,6% | 0,03125 |
| **Bard-Chain-of-thought** | 4 | 2 | 2 | 58 | 50% | 3,3% | 0,0625 |
| **GPT-3,5-One-shoot** | 5 | 4 | 1 | 56 | 80% | 6,6% | 0,12308 |
| **GPT-3,5-Chain-of-Thought** | 0 | 0 | 0 | 60 | 0 | 0 | 0 |
| **GPT-4-One-shoot** | 7 | 4 | 3 | 56 | 57% | 6,6% | 0,1194 |
| **GPT-4-Chain-of-Thought** | 3 | 2 | 1 | 58 | 66% | 3,3% | 0,06349 |
| **Llama2-One-shoot** | - | - | - | - | - | - | - |
| **Llama2-Chain-of-thought** | - | - | - | - | - | - | - |

***PHASE 1.5***

Τhis phase of the experiments was performed only based on our question of whether LLMs will be able to process SAR documents and extract knowledge and information to be used in the formation of ontologies. Essentially it only concerns GPT4 which has the ability to upload and edit files. The uploaded file was a blank forest fire document used by the Greek Fire Department to record fire incidents of any kind. The results were impressive, given the model's limited period of operation (generally available to users July 2023), but also its use by users, as it is a paid model and not many users choose it.

The template was to first introduce the document sheet and ask GPT-4 to start engineering and ontology. With fine-tuning and after 10 prompts the results where interesting. We can observe a progress as far as it concerns the metrics, maybe it needs a refinish, but for sure as long as we continue fine-tuning, the result is better and the model has an improved performance.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No of Prompts [[3]](#footnote-3) | Results | | | | | | | |
| Axiom | Logical axiom count | Declaration axioms count | Class count | Object property count | Data property count | Individual count | Annotation property count |
| 1 | 15 | 1 | 1 | 8 | 0 | 0 | 1 | 6 |
| 2 | 28 | 2 | 14 | 14 | 0 | 0 | 2 | 6 |
| 3 | 33 | 3 | 15 | 15 | 0 | 0 | 3 | 9 |
| 4 | 33 | 3 | 15 | 15 | 0 | 0 | 3 | 9 |
| 5 | 33 | 3 | 15 | 15 | 0 | 0 | 3 | 9 |
| 6 | 83 | 3 | 15 | 15 | 0 | 0 | 3 | 18 |
| 7 | 10 | 0 | 10 | 10 | 0 | 0 | 0 | 0 |
| 8 | 122 | 0 | 10 | 10 | 0 | 0 | 0 | 20 |
| 9 | 141 | 12 | 17 | 11 | 6 | 0 | 0 | 20 |
| 10 | 171 | 112 | 27 | 11 | 6 | 10 | 10 | 12 |
| 11 | 182 | 121 | 31 | 14 | 9 | 10 | 10 | 10 |

***PHASE 2***

This is the most crucial moment of the experiments. We have to define and construct a well-thought prompt engineering in order to succeed the most productive and best outcomes and take advantage of all the features and capabilities that every model has to offer. As ontology engineers but simultaneously as domain experts in the field of SAR in our team, we have reached to a series of prompts that later were able to fine-tune and alter at specific and crucial points. Faithfully following the proposed X-HCOME methodology, the set of prompts that were used in this phase is depicted below:

1. Act as an ontology engineer. Your task is to build an ontology based on the data and info that I will provide you in the following prompts. You will not start generating the ontology until the moment that I will ask you. You must take in account and consider all the requirements and specifications that the ontology must meet and then produce it in the format that I will ask.
2. The aim of the ontology that you will create below is to provide a framework for data representation and interlinking of wildfire events, fostering advanced reasoning, situational awareness, and interpretation for decision support in forest fire emergencies, especially in search and rescue missions. The scope of the ontology is to symbolically model contextual information in the domain, addressing application and user requirements and promoting the creation of interoperable knowledge graphs.
3. In terms of the depth or scope of domain knowledge to be represented, the ontology that you will build must aim to be comprehensive, covering all aspects of forest fire management across the three different phases of wildfire management: prevention & preparedness, detection & response, and adaptation & restoration. Your design must capture a wide range of relevant incidents and impacts in a wildfire disaster, associated weather conditions, data from human and earth observations, missions, and relationships between the services. It must also meet the main requirements that a Search and Rescue Mission in wildfire events has to cover, in order to be documented in detail and be used for future analysis. Therefore, the generated ontology will be intended to have a deep scope, encompassing a broad range of domain knowledge relevant to forest fire emergencies.
4. The key knowledge that must be represented in your ontology includes:

A. Incidents and Impacts: The ontology must capture relevant incidents and impacts in a wildfire disaster, such as the spread of the fire, damage to infrastructure, and ecological impacts. This knowledge is crucial for understanding the extent and severity of the wildfire and its consequences.

B. Weather Conditions: Representation of weather conditions, including temperature, wind speed, humidity, and weather forecasts, is essential for understanding the environmental factors influencing the behavior of the wildfire. This knowledge helps in assessing the potential spread and behavior of the fire.

C. Data from Human and Earth Observations: The ontology must include data relevant to the analysis of input data coming from various sensors, satellites, and social media sources. This knowledge provides valuable information for monitoring and assessing the wildfire situation.

D. Missions and Relationships Between Services: Representation of missions and relationships between the services involved in wildfire management is important for coordinating and organizing emergency response efforts. This knowledge facilitates effective decision-making and resource allocation.

E. Search and Rescue Team: The ontology must include all the personnel and equipment that are involved in a SAR mission. From rescue to medical support team (not only humans but also search animals), special forces (for example if army/airforce is engaged in the mission etc.), technological equipment for searching, as well as other aspects of a SAR mission that need to be documented.

The representation of this knowledge is essential for facilitating advanced reasoning, situational awareness, and decision support in forest fire emergencies. By capturing and interlinking this knowledge, the ontology enables a comprehensive understanding of the wildfire situation, supports informed decision-making, and promotes the creation of interoperable knowledge graphs for effective crisis management.

1. In this phase you will be provided with Documents from the Fire Department with real-case forest fire incidents and real data. You will examine these data and you will incorporate this knowledge from these files into your ontology and adapt all of these information, in order to make the ontology more efficient and inclusive. In case that you have already considered these information (cross-information), or the meaning is similar, don’t double record/register.
2. I will provide you with three sets of Competency Questions (CQs). Competency Questions are a set of Natural Language Questions that must be answered correctly by the ontology that you will generate and they are crucial in the ontology development process, since they represent ontology needs. You must take into account these CQs and your ontology must answer them. Here is the first set that related to representation to wildfire disaster and relevant incident and impacts:

CQ1. What are the most important weather variables that can cause forest fire?

CQ2. What are the current measurements for these weather variables?

CQ3. What is the forecast for the weather in this location?

CQ4. Where did the incident take place?

CQ5. What is the priority of an incident during a forest fire disaster?

CQ6. What incidents during forest fires are the most urgent?

1. The next set of CQs that you must take into account is related to representation of data from human and earth observations:

CQ7. What data from the source are depicted?

CQ8. Which is the creation date of these data?

CQ9. What is the location of this item?

CQ10. Which is the classification type of smoke?

CQ11. Which vulnerable objects were involved in the incident?

CQ12. What is the status of wildfire forestry works (firebreaks, access to forest roads, etc.)?

1. The last set of CQs that your ontology must be able to answer is related to representation of missions and relationships between the services:

CQ13. What services or support do you offer for firefighting?

CQ14. Which mission do you follow for this support/service?

CQ15. What is the location where this mission is taking place?

CQ16. Where is the most urgent mission taking place?

CQ17. What is the population density in the area?

CQ18. What is the location of the involved people?

1. Search for other ontologies in the same-similar domain and adapt crucial points on your ontology engineering. Take in mind SoKNOS and BeWARE ontology, two examples o emergency management ontologies that can be helpful in the “Search and Rescue Mission” part of the ontology that you will create. Take into serious account the term of reusability other ontologies because it can contribute on your outcome in a successful way.
2. Now develop/build/create the ontology based on all the above information from the prompts that I asked you. You must take into account all of the information and knowledge that you have produced since the first prompt of our conversation. Be careful in the representing the knowledge, meeting all the specifications that I’ve already asked you. Act not only as an ontology engineer but also as domain expert in Search and Rescue in wildfire incidents.The output file must be in .ttl format and must be opened in Protégé 5.6.3.

We run this set of prompts in all 4 LLMs, but in order to reach the best outcome possible, we wanted to ensure and evaluate the produced ontology, by engineering prompts that should confirm that it meets the expectations and specifications given during the phase.

Based on these prompts, the results that emerged are documented in the following tables with the relative comments and analysis.

**Mapping**

CLASSES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Classes** | **True Positives** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 80 |  |  |  |  |  |  |
| **GPT4-One-shoot** | 8 | 6 | 2 | 74 | 75% | 7,5% | 0,136 |
| **GPT4-Chain-of-Thought** | 5 | 4 | 1 | 76 | 80% | 5% | 0,094 |
| **XHCOME** | 17 | 15 | 2 | 65 | 88% | 18,75% | 0,3092 |

OBJECT PROPERTIES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Obj. properties** | **True Positive** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 60 |  |  |  |  |  |  |
| **GPT4-One-shoot** | 7 | 4 | 3 | 56 | 57% | 6,66% | 0,1194 |
| **GPT4-Chain-of-Thought** | 3 | 2 | 1 | 58 | 66,6% | 3,33% | 0,0634 |
| **XHCOME** | 12 | 10 | 2 | 50 | 83,33% | 16,66% | 0,2777 |

Satisfactory and efficient performance as far as it concerns ontology engineering. A huge positive attribute/benefit is that capability to generate the final file for download without needed to be copied in a text editor and the opened with Protégé. Had the ability to process the attached files and produce the needed knowledge. Maybe it would give better results if at the stage of prompt no5 before the documents uploading, should generate the first draft of the ontology and then build up with the extra information given to better and more inclusive versions of the ontology. Huge disadvantage: the limit of messages during time. For a subscription of 25€ shouldn’t limit to 40 messages/3 hours. Had to stop our experiments many times and execute them from scratch. Ιn general terms a good choice, if you seek for convenience as far as it concerns the functionality of the model and you need to edit files and documents.

Chat GPT-3.5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No of Prompts | Results | | | | | | | |
| Axiom | Logical axiom count | Declaration axioms count | Class count | Object property count | Data property count | Individual count | Annotation property count |
| 1-11 | 81 | 53 | 28 | 11 | 11 | 9 | 7 | 0 |
| After fine-tuning + extra prompts for boosting the ontology | 655 | 306 | 59 | 30 | 21 | 19 | 102 | 31 |

**Mapping**

CLASSES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Classes** | **True Positives** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 80 |  |  |  |  |  |  |
| **GPT-3,5-One-shoot** | 6 | 4 | 2 | 76 | 66% | 5% | 0,093 |
| **GPT-3,5-Chain-of-Thought** | 14 | 7 | 7 | 73 | 50% | 8.75% | 0,148 |
| **XHCOME** | 30 | 22 | 8 | 58 | 73% | 27,5% | 0,4 |

OBJECT PROPERTIES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Obj. properties** | **True Positive** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 60 |  |  |  |  |  |  |
| **GPT-3,5-One-shoot** | 5 | 4 | 1 | 56 | 80% | 66% | 0,12 |
| **GPT-3,5-Chain-of-Thought** | 0 | 0 | 0 | 60 | 0 | 0 | 0 |
| **XHCOME** | 21 | 16 | 5 | 44 | 76% | 26% | 0,39 |

Probably the one that conquers the top, alongside with Bard, but each of them for a different reason. Because of the most-being trained time of them all, it generates faster and more productively the needed results from all of them. If it had also the ability to process files attached it would be much better, needed to convert in text-natural language all the document files in order to process them and extract the knowledge needed on order to build the ontology, or parts of it (especially the examples). Sometimes, during the series of experiments repeated in the procedure, it seemed to forget the provided knowledge, so had to remind it on all the next prompts the memo to re-consider all the above knowledge from prompt 1. Gave the most axioms of all of them as long as individuals, maybe not in the wanted form, but with alternations from the user, it reaches to an excellent point (we are talking about a hybrid model so it is legitimate).

BARD

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No of Prompts | Results | | | | | | | |
| Axiom | Logical axiom count | Declaration axioms count | Class count | Object property count | Data property count | Individual count | Annotation property count |
| 1-11 | 86 | 53 | 32 | 23 | 8 | 6 | 10 | 1 |
| After fine-tuning + extra prompts for boosting the ontology | 169 | 111 | 58 | 32 | 16 | 16 | 18 | 0 |

**Mapping**

CLASSES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Classes** | **True Positives** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 80 |  |  |  |  |  |  |
| **BARD-One-shoot** | 19 | 8 | 11 | 72 | 42% | 10% | 0,1616 |
| **BARD-Chain-of-Thought** | 8 | 4 | 4 | 76 | 50% | 5% | 0,0909 |
| **XHCOME** | 32 | 27 | 5 | 53 | 84% | 34,5% | 0,4821 |

OBJECT PROPERTIES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Number of Obj. properties** | **True Positive** | **False Positives** | **False negatives** | **PRECISION** | **RECALL** | **F-1 SCORE** |
| **Gold** | 60 |  |  |  |  |  |  |
| **BARD-One-shoot** | 4 | 1 | 3 | 59 | 25% | 1,66% | 0,013 |
| **BARD-Chain-of-Thought** | 4 | 2 | 2 | 58 | 50% | 3,33% | 0,062 |
| **XHCOME** | 16 | 14 | 2 | 46 | 87,5% | 23,33% | 0,368 |

As noted before, it reaches the top alongside with GPT-3.5. It meets a bigger response-time in the matter of generating the results of the provided prompts. The only one of the 4 that gave by itself a SWRL rule, but still couldn’t be edited in protégé so had to be deleted from the final code. Probably every ontology should be converted in OWL so that the rules could be editable. Often stopped the procedure of generating the code in half, so had to ask it to continue generating, if the code was big, perhaps 3-4 times. The generated codes had occasionally syntactical errors, so had to fix them in order to continue the experiments. Understood completely the given knowledge, but it also sometimes couldn’t follow the sequence of events (as GPT-3.5) and skipped info that had already generated from the provided knowledge. In terms of knowledge and content slightly better from GPTs, but worse in matters of interface and functionality.

LLAMA2

All the generated ontologies couldn’t be opened/edited with protégé 5.6.3. We ran the prompt sets many times, all of them couldn’t really respond to the specifications needed, not recommended for ontology engineering, many times stopped the procedure and needed to be executed from the start, the generated code is neither in turtle syntax nor in any form of ontology coding.

1. All the ontologies saved in this repository <https://github.com/dimitrisdoumanas19/New-Experiments-LLMs.git> [↑](#footnote-ref-1)
2. Panagiota Masa, Georgios Meditskos, Spyridon Kintzios, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2022. Ontology-based Modelling and Reasoning for Forest Fire Emergencies in Resilient Societies. In Proceedings of the 12th Hellenic Conference on Artificial Intelligence (SETN '22). Association for Computing Machinery, New York, NY, USA, Article 24, 1–9. <https://doi.org/10.1145/3549737.3549765> [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)