UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Learning based on user interaction.

Bachelor Thesis

Arnau Martí Llobet

Director: Javier Vázquez Salceda
June 23, 2021

Contents

1 Context and scope **8**

1.1 Introduction and contextualization . . . 8

1.1.1 Context . . . 9

1.1.2 Concepts . . . 10

1.1.3 Problems to be solved . . . 12

1.1.4 Stakeholders . . . 13

1.2 Justification . . . 13

1.2.1 Previous studies . . . 13

1.2.2 The need of a Collaborative AI System . . . 14

1.3 Scope . . . 15

1.3.1 Objectives and sub-objectives . . . 15

1.3.2 Requirements . . . 16

1.3.3 Potential obstacles and risks . . . 16

2 Methodology and project planning **18**

2.1 Methodology and rigor . . . 18

2.1.1 Methodology . . . 18

2.1.2 Tools and validation . . . 19

2.2 Description of the tasks . . . 19

2.2.1 Task definition . . . 19

2.2.2 Summary of the tasks . . . 22

2.2.3 Resources 22
2.3 Initial Gantt 23
2.4 Risk management 23
2.4.1 Inability to adapt existing components and tools 24
2.4.2 Inexperience in the field 25
2.4.3 Ongoing platform 25
2.4.4 External factors 25
2.5 Final work plan 26

3 Design process **28**
3.1 Early Design and changes 28
3.1.1 The NLP engine 28
3.1.2 The Knowledge graph 29
3.2 Early Prototyping 31
3.3 Final Design 32
3.3.1 NLP Pre-processing (FreeLing + Query generation) 32
3.3.2 Knowledge Graph Management (Knowledge Reasoner & Manager + Neo4j) 33
3.3.3 Visualization (Front-end + Back-end) 33

4 Implementation **35**
4.1 Starting point: existing tools and components 36
4.1.1 FreeLing 36
4.1.2 Neo4j 37
4.1.3 NodeJS 37
4.2 Developed components 38
4.2.1 Dependency processing (Query Generation module) 38
4.2.2 Expansion of the concept model (Neo4j database) 39
4.2.3 Graph views (Front-end + Back-end modules) 41
4.3 Implemented algorithms (Knowledge Reasoning & Management module) 42

4.3.1 First semantic method . . . 43

4.3.2 Second semantic method . . . 43

4.3.3 First similarity method . . . 45

4.3.4 Second similarity method . . . 45

4.3.5 Centrality methods . . . 46

4.3.6 Community detection method . . . 47

4.3.7 Link prediction method . . . 47

4.3.8 Node similarity method . . . 48

5 Testing **49**

5.1 Real-world data . . . 49

5.2 Result analysis . . . 50

5.2.1 Topic and category recognition . . . 50

5.2.2 Suggestion evaluation . . . 52

6 Conclusions and future work **56**

A Source code **58**

B Integration of knowledge (Technical competences) **58**

C Identification of laws and regulations **59**

D Budget **61**

D.1 Staff costs . . . 61

D.2 Generic costs . . . 62

D.3 Budget deviations . . . 63

D.4 Management control . . . 64

E Sustainability **65**

E.1 Self assessment . . . 65

E.2 Environmental . . . 66

E.3 Economic 66
E.3.1 Social 67

F Output queries (extract) 68

G Output suggestions 71

List of Figures

1-1 Knowledge Graph example [2] 9

2-1 Gantt chart, made using TeamGantt 24
2-2 Updated Gantt chart, made using TeamGantt 27

3-1 Design of the proposed architecture 32
3-2 Introductory view of the classifier 34

4-1 Architecture of the implemented system. 35
4-2 Dynamic suggestions plus new topic addition form (included with auto-complete function) 38
4-3 AI Resource node related to both topics and categories 40
4-4 Explore view category filters 41
4-5 Neovis graph with new topic 42
4-6 Second semantic output Neo4j graph 44

5-1 Multiple topics being found on a document images classification resource 51
5-2 Compound of terms being recognized as a topic 51
5-3 Business category suggestions 51
5-4 Technical category suggestions 52
5-5 `"motion sensor"` recognized via the second semantic method 54

List of Tables

2.1 Summary of the tasks and subtasks 22

D.1 Cost per hour of the different roles [39]. 61

D.2 CPA Table 62

D.3 Total of the generic costs table 63

D.4 Incidental costs table 63

D.5 Final budget table 64

Chapter 1

Context and scope

1.1 Introduction and contextualization

Recently, the Knowledge Graph model has been widely adopted in conjunction with established Semantic Web technologies to support the need of representing knowledge in a more insightful manner. Solving the problem of integrating new entities and their relationships while growing seamlessly, and thus enabling the ability to make supported inferences dynamically.

The "Knowledge Graph" term and even its popularity have been influenced by the introduction of Google's Knowledge Graph [1] in 2012, for its use in the improvement of their search engine. As a perfect fit for these types of use cases where there is a sheer amount of (new and old) information, to make sense of, we can use those graphs to match with user-provided information in order to make recommendations.

In this thesis, we are focusing on the latter, the human-assisted feedback of an existing and, potentially, evolving knowledge graph. Using the user interaction and combining it with domain analysis will expand its representation.

The goal is to create a prototype system that explores the use of a combination of Ontologies and Knowledge Graphs to extend the capabilities of the AI4EU platform. AI4EU [3], aims to constitute a one-stop-shop for European AI, enabling end-users to discover, learn and upload AI resources for its use in everyday processes. This means

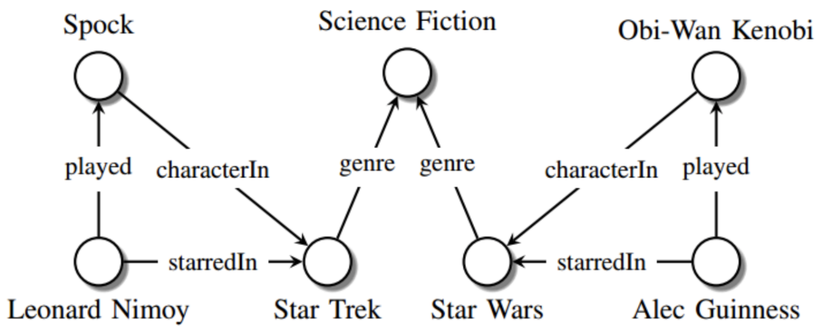


Figure 1-1: Knowledge Graph example [2]

categorization and adaptability of new concepts can be key. Currently, the use of semantic technologies is already extensively used to search the resources.

The intention is to improve the workflow of adding new AI resources, not only by helping the user in the selection of existing terms, but also by allowing the user to propose new concepts and relations to the static knowledge graph. When a user publishes a resource and by only adding a text description, the system would have to segment it via natural language recognition and be able to, using the knowledge graph, auto-fill the tags and categorize its features. This first part helps the user have a better experience and not having to describe the resource as 'other'. The other improvement is made in the back-end, suggesting new forms of entities and relationships as needed for the Knowledge Graph (Figure 1-1) to grow into, adapting it to the user's new work and not limiting it to pre-established terms.

1.1.1 Context

This is a Bachelor Thesis (Degree Final Project) of the Bachelor Degree in Informatics Engineering, within the Computing specialization, done at the Facultat d'Informàtica de Barcelona of the Universitat Politècnica de Catalunya - BarcelonaTech and directed by Javier Vázquez Salceda.

1.1.2 Concepts

This section consists of an introduction to the main concepts that are used in the description of the project.

Natural language processing

The natural language recognition part of the system serves up as a link between the user's text input and the machine-ready semantic graph used by the system. Making sense of and extracting significance from language is opening up the possibility to build a partial knowledge graph, that then can be used to search and query databases.

FreeLing [4] is a tool that will enable us to do that. The project was created and is currently led by Lluís Padró, as a result of the work done by the Natural Language Processing research group (NLP) which is part of the Center for Language and Speech Technologies and Applications (TALP) at UPC. Built in C++, the library provides a diverse set of language analysis and process functionalities such as morphological analysis, named entity detection, PoS-tagging, parsing, Word Sense Disambiguation, Semantic Role Labelling, etc. for multiple languages including Catalan and Spanish but, most importantly for this project, English.

In our case, we will focus on Freeling's capability to make sense of whole paragraphs in the AI resource description and then generate a Semantic Graph. This will be one of the important inputs used by the system. We will explore if adding extra, AI-related ontologies to Freeling may increase its capability to recognize relevant features in the user-provided descriptions.

Conceptual Semantic Model

The Ontology designed for AI4EU [5] acts up as a form of a Conceptual Semantic Model (also called Concept Graph), which then guides the evolution of the Knowledge graph that is populated at run-time. An ontology usually is fixed in terms of classes and relations and does not vary in execution time. In the same way, the knowledge graph in the actual platform is being restricted in terms of growth and cannot surpass the limits defined by the ontology, and therefore it can only save instances of existing classes and relationships. But in this project, we will explore the possibility not only for the Knowledge graph to include new classes and relations, but also to add those back to the ontology, either directly or on a case-by-case basis.

Describing the AI resource requires the concept model design to part from broad concepts in its top sub-classes. For example, starting from the ones that conceptualize different types of technology in a resource e.g. Dataset, Software, Hardware, Ontology, etc. Also, multiple Distributions of the same resource can be specified, and attachments in the form of Documentation to each one such as wikis, user manuals, or code examples. Moreover, the characterization of a resource is made through descriptive entities like technical or business topics it touches, relevant keywords, or the type of assets that it relies on. These last couple are the most prone to require inclusion/modifications.

The definition of the AI4EU model has been made following the Semantic Web concepts and standards set by the W3C (World Wide Web Consortium) to make Internet data machine-readable. The Ontology follows the OWL (Ontology Web Language) specification, and it extends existing ontologies such as the Dublin Core Terms [6] (for industry-standard constructs and vocabulary), the FOAF [7] (defining basic resource, person and organization information) and the Computer Science Ontology [8] (describing broadly the Computer Science domains, concepts, and terminology). A pair of new controlled vocabularies were designed exclusively for the project, *Research Areas* and *Application Areas*, which are static lists of tags that are used to describe the AI resources in the AI4EU platform.

The conceptual schema has relations defined between all the core entities, so a query will be able to start from any class and navigate through it. Extending and maintaining the semantic model is done through a series of tools and validations. First, Protégé is used for editing the ontology and then via GitHub changes are submitted. Then, requests and collaboration follow until a major update has to be approved by the AI4EU WP3 task leader.

Knowledge Graph

The knowledge graph is based on the preceding ontology and is the foundation that allows the description of the resources and at the same time allowing the search and visualization mechanisms of the AI4EU platform [5].

The graph is specified using RDF following the definition made by the W3C. The RDF graph is a labeled oriented multi-graph made of edges, which are known as triples, and nodes. The nodes represent resources while the edges or triples of the graph represent their properties. RDF resources are assigned a type using the classes which can be defined using RDFS.

The deployment of the AI4EU graph is done using a triple store Semantic Web tool. A triple store is a graph database that implements RDF, RDFS, and the SPARQL query language. This means that the triple store provides a SPARQL endpoint, which makes the Knowledge Graph accessible via an HTTP protocol, therefore it can be queried and navigated. There are various ways of loading in new data, one of those is uploading directly RDF metadata or simply documents loaded with SPARQL update statements. The resource searches can also be done in SPARQL query language.

AI4EU provides a website that provides visual representations of the server for doing queries using relevant keywords. Additionally, it provides an RDF validation service, using the defined SHACL shapes, both for validating the graph, in RDF form, and the ontology.

The knowledge graph and an overlying search and visualization service was built and deployed, showcasing the possibilities of using the knowledge graph for complex querying and inferring. Implementing logical rules either extracting it from the graph or adding it is a way of improving the knowledge representation. In addition, the explainability of the results would be backed up by the nodes of the graph.

1.1.3 Problems to be solved

Even though the AI4EU platform improves upon other platforms, and incorporates promising model resources, tools and datasets, there is room for extension and improvement in both the Conceptual model and the user experience. In this thesis, we want to make use of the knowledge graph capabilities and add new functionalities to the platform that enhance the AI resource categorization process while seamlessly getting a better user experience. The aim is to assist the user when he/she is publishing resources and help in its classification. Furthermore, the system will extend its classification capabilities by learning new categories from interaction with the users, and this not only may help the addition of other similar resources but the search of interested users on the newly defined categories.

1.1.4 Stakeholders

A few parties are involved in this project, directly or indirectly, depending on the type of participation in the work. The director, Javier Vázquez Salceda, a member of the KEMLg group within UPC's IDEAI research center (which is one of the partners in the AI4EU project), and the student, are stakeholders that have a direct implication on the thesis.

Indirectly implicated in the thesis are the ones that may be affected by the work. On one side the AI4EU project might integrate the proposed learning system in their public website and service, and it will encourage other platforms within the community to consider using these methods. On the other side are the users of the AI4EU platform, that might benefit from the technology proposed in this work.

1.2 Justification

1.2.1 Previous studies

In the phase of definition of the AI4EU conceptual models [5], the researchers analyzed the data models of other big European initiatives such as three Horizon2020 Eu-funded projects: BEAT [9], BONSEYES [10], and ELG [11], which are sort of "competing" platforms. For example, BEAT offers the possibility of combining multiple AI resources. During the development, it was decided in favor of platform inter-compatibility with also including a built-in ELG mapping layer. With feature parity and already support for these multiple databases the backlog of data for the graph is solved, hence there is room for incorporating better learning with this and new information.

The usage of proprietary knowledge graphs has been implemented by big and small companies that focus on better understand their users' data with an evolving and malleable scheme. The likes of Netflix [12], saw a good fit for a recommender system that with every watch both a singular profile and the community knowledge changes and is learned. First to be destined for the Semantic Web, sectors from machine learning and researchers are focusing on the reasoning and inference side of it, solving Data Insufficiency, Zero-Shot Learning and Explainability [13]. Improvements to complex queries are being made to resolve Five W's searches [14], and even incorporating a Computer Vision supported by a learning graph system-controlled robot [15].

1.2.2 The need of a Collaborative AI System

As seen in the previous section, the progress that can be made with the learning knowledge graph based on the user assistance is something that the AI4EU platform has the elements in place to have incorporated into. Moreover, this is the kind of service that requires a good match of what a resource really is about, to represent it, and to align it with the users' provided description.

There is a need of a Collaborative AI system (i.e., a system where humans and AI collaborate to better perform in a task. In this case we have both directions present in a Collaborative AI system:

- *the AI helps the user*: The current workflow for resource publishing is a hard, time-intensive task for users, specially non-experts. This could end up in less users willing to share their resources and to use the platform. The proposed system in this work could enhance the user experience with the assistance of an AI system to ease the registration of AI resources in the platform.

- *the user helps the AI*: by suggesting new terms to better classify Ai resources, the user is driving the learning process of the AI system.

Furthermore, the user can help the AI in rod

1.3 Scope

The objectives, requirements and methodology set on the following sections are backed with the planning stated on §1.

1.3.1 Objectives and sub-objectives

General Objective Description: The system should analyze the textual description of the AI resource provided by the user and assist him/her in the classification process, identifying new terms and relations not present in the existing Conceptual Model.

To accomplish the objective, we can subdivide the project into numerous sub-objectives:

1. **Extracting terms from the AI Resource's text description and mapping them to the Ontology**

    - Study how to integrate the AI4EU ontologies into the FreeLing analyzer.
    - Implement an adapter to convert Freeling's Semantic Graph output into a Knowledge Graph compatible with AI4EU's Knowledge Graph.

2. **Matching the AI Resource terms with the AI4EU Ontology**

    - Create an ontological reasoning module to do the matching between the Knowledge Graph extracted from the AI Resource Description and the AI4EU Ontology and Knowledge Graph. Exact matches will become classification recommendations to the user. Mismatches will be the source for the Ontology learning alongside the dynamic knowledge graph from the data, which is done in the next objective.

3. **Creating an interactive method to learn new terms and relations with the help of the user**

    - Using the mismatches found in the ontological reasoning module as input, develop an interactive AI system where the user helps in the selection of terms and relations that are a candidate to be added to the AI4EU Ontology.

1.3.2 Requirements

Some requirements are needed to ensure the quality of the project:

- The prototype should have an intuitive, easy-to-use interface.
- The prototype should have a modular architecture where components have clear interfaces, to ease the connection to other components or the substitution of the proposed components with future ones.
- The algorithms developed in the project should be correctly written, tested and documented, to ease future extensions.
- The system should be able to recognize significant categorical features of a resource from its textual description.
- The data models (Ontology and Knowledge Graph) should be adaptable to new knowledge shifts and changes over time.

1.3.3 Potential obstacles and risks

Throughout the development of the thesis, some potential risks and obstacles that may show up and may have to overcome.

- Inability to adapt existing components and tools: Certain tools and components may be hard to extend or connect and we will not be able to support some features integrated directly, requiring more effort to implement them separately. Also, the unavailability of the platform's source code can add delay.

- Inexperience in the field: It is the first time I'm getting involved in a project of these characteristics involving other parties and with topics that I'm not fully familiarized with.

- Ongoing platform: The AI4EU platform is an already established service. If there are relevant changes in the AI4EU Data Models these may impose changes in our models during the development of the thesis.

- External factors: Working on a thesis amid a global pandemic can reduce or slow down certain processes and interactions. There is even a risk to go back to full confinement or to harder social distancing measures. Inevitably some of these conditions may affect the execution of the thesis. This risk will be handled by the continuous monitoring of the project's evolution, and the ability to change priorities or replan during the project lifetime.

Chapter 2

Methodology and project planning

2.1 Methodology and rigor

Defining adequately the working methodology that is going to be followed, the monitoring tools, and the validation methods is decisive to the correct development of the thesis.

2.1.1 Methodology

Given that the thesis has to be completed in a short period of time (in about 4 months) and due to its exploratory, research nature, we have opted for an agile methodology that subdivides the whole work into smaller tasks called sprints. As the project does not have lots of little objectives and instead relies on bigger proof of concepts, we have done two-week sprints. At the beginning, the focus has been on creating basic working prototypes of the natural language part, and then following on to the standard ontology adaptation, to keep going down to create as soon as possible, a prototype that connects a basic version of all components. Each sprint has consisted of working towards an essential feature to its bare-bones state and then jumping on to the next until we reach the extension phase when it will consist of improving upon the existing. Most of the time has been concentrated on reaching a good design that we are comfortable implementing to avoid wasted code and accomplish faster developing times.

2.1.2 Tools and validation

We are using the GitHub repository as a version control tool because it is easy to access by everyone involved and also offers data resilience. The code will be publicly available to the parties involved, so they all have the possibility to follow the work of the prototype and the progress made. The Ontology and other provided data by the AI4EU platform will allow an easy and real validation of the project results. Task execution was meant to be tracked by means of Trello, but the timely weekly meetings made it a non-requirement. The meeting is scheduled with the director of the project, with the aim of discussing the project status and planning the tasks to be completed in the next weeks.

2.2 Description of the tasks

The total amount of hours committed to the project will be approximately 540 hours, which is the equivalent of 30 hours for every credit of a total of 18. The begin date is on 23rd of February and the delivery date on the 21st of June, so the 540 hours of work are distributed among 119 days. As the intention is to present the thesis within the first call (from 28th June to 2nd July), we have foreseen a week-long gap to deliver the final document to the committee in advance, and to prepare the oral presentation.

2.2.1 Task definition

This section shows the tasks that will be done throughout the project. Additionally, each task is described taking into account the dependencies with other tasks and the duration of each one. We assembled the tasks in groups that are ordered accordingly to their part in the project structure. Therefore, there may be dependencies between each task, or otherwise groups that can be done independently. For example, with the purpose of beginning to work on the practical implementations we need to first finish the preliminary study, even though each development can be independent of the others.

1. **Project planning and documentation**

Given the importance of the planning to get the project on track, in this task we group the subtasks that will be done during project management, with the intention of documenting and defining the work to do.

- Contextualization and project scope: Definition of the project scope and a contextualization of its field. Setting the objective of the thesis, the justification of the project and how it is going to be developed.
- Temporal planning: Organization of the upcoming work, with a brief description of the task and subtasks, resources and requirements needed.
- Economic management and sustainability: Analysis of the economic and the sustainability side of the project.
- Documentation: Started during the project planning and made throughout the duration of the whole project. It is important to document the work done periodically until the date to avoid the doing it all in the final weeks.
- Preparation of the oral defense: the presentation will consist on a distilled version made of the important parts of the thesis worth explaining.
- Meetings: Reunions with the tutor will be scheduled every week with the purpose of ensuring that the thesis is progressing correctly and pursuing the time plan set.

2. **Preliminary work**

   Considering that one of the main purposes of the thesis is to implement a project that includes several different parts and technologies, it is necessary to have an understanding of the inner-workings of each tool/platform used. FreeLing and both AI4EU's Ontology and Knowledge Graph manager (Corese [16]) are the tools being evaluated.

3. **Development of the Partial Knowledge Graph**

   The first problem we are going to solve is how the data will be extracted from a description text file. Beginning from inspecting the possibilities of the Natural

Language recognition tool chosen (FreeLing), the goal is to elaborate a strategy to incorporate the AI4EU ontology into it. As the software will be producing a semantic graph, we will need to implement a knowledge graph conversion tool, while also checking the degree of compatibility the graph has with the AI4EU schemes.

4. **Development of the Reasoning Module**

   Next, we will plan a reasoning method to incorporate the recognized language into the existing ontology and graph, which will also needs to be analyzed. Implementing a combination module will have to differentiate what knowledge is already in the ontology and which is new and relevant to learn. In the course of the development, we will look into the correctness of the matching decisions and the accuracy of the proposals of new terms and relations.

5. **Development of the Learning System**

   This part will consist in defining how to interact with the user when incorporating the new terms. Developing an interactive system to assist the user with the definition of the resource, while generating the new graph and expanding the ontology. Then, we will verify if the learning from the user's interaction is transferred appropriately into the ontology.

   As seen in this and previous sections, each development is made of three differentiated parts: design, implementation and validation.

6. **Integration and final evaluation**

   In the final part of the development side, the task is to integrate the previous blocks into a whole system, and then evaluate it for the first time together. Using tests to convey the improvement that it would bring to the platform in learning capabilities.

2.2.2 Summary of the tasks

In the following table 2.1 there are summarized the previously explained tasks, showing the dependencies among them and with the amount of hours dedicated to each one.

Id.	**Task**	**Time (h)**	**Dependencies**
T1	**Project planning and documentation**	110	
T1.1	- Contextualization and project scope	24	
T1.2	- Temporal planning	8	T1.1
T1.3	- Economic management and sustainability	8	T1.1, T1.2
T1.4	- Documentation	60	
T1.5	- Oral defense preparation	10	T1.4
T1.6	- Meetings	20	
T2	**Preliminary work**	30	
T2.1	- FreeLing	10	
T2.2	- Ontology	10	
T2.3	- Corese	10	
T3	**Dev. of the Partial Knowledge Graph**	110	T2.1
T3.1	- Design	12	
T3.2	- Implementation	90	T3.1
T3.3	- Validation	8	
T4	**Development of the Reasoning Module**	110	T2.2
T4.1	- Design	12	
T4.2	- Implementation	90	T4.1
T4.3	- Validation	8	
T5	**Development of the Learning System**	110	T2.3
T5.1	- Design	12	
T5.2	- Implementation	90	T5.1
T5.3	- Validation	8	
T6	**Integration and final evaluation**	50	T4, T5, T6
T6.1	- Integration	30	
T6.2	- Final evaluation	20	

Table 2.1: Summary of the tasks and subtasks

2.2.3 Resources

To help and carry the work made on the thesis we will make use of some, human and material, resources that are important to mention.

Human resources

The main human resource of the thesis is the researcher, therefore it is who will carry the work forward. The thesis director Javier will mentor the researcher on the task, so it is considered a necessary human resource. The GEP tutor has the task of correcting the project management part.

Material resources

- Overleaf: We will use Latex given the easy text formatting and organization, and Overleaf is the best online tool for it.
- IDE: A complete IDE is going to be needed in order to make more fluid the development.
- Protégé: An established ontology editor is required to visualize both the AI4EU concept model and graph.
- Trello: To better micro manage the tasks/subtasks and allow the director to follow the progress an organization tool will be used.
- Github: Will be used to store the code and as a version control utility.

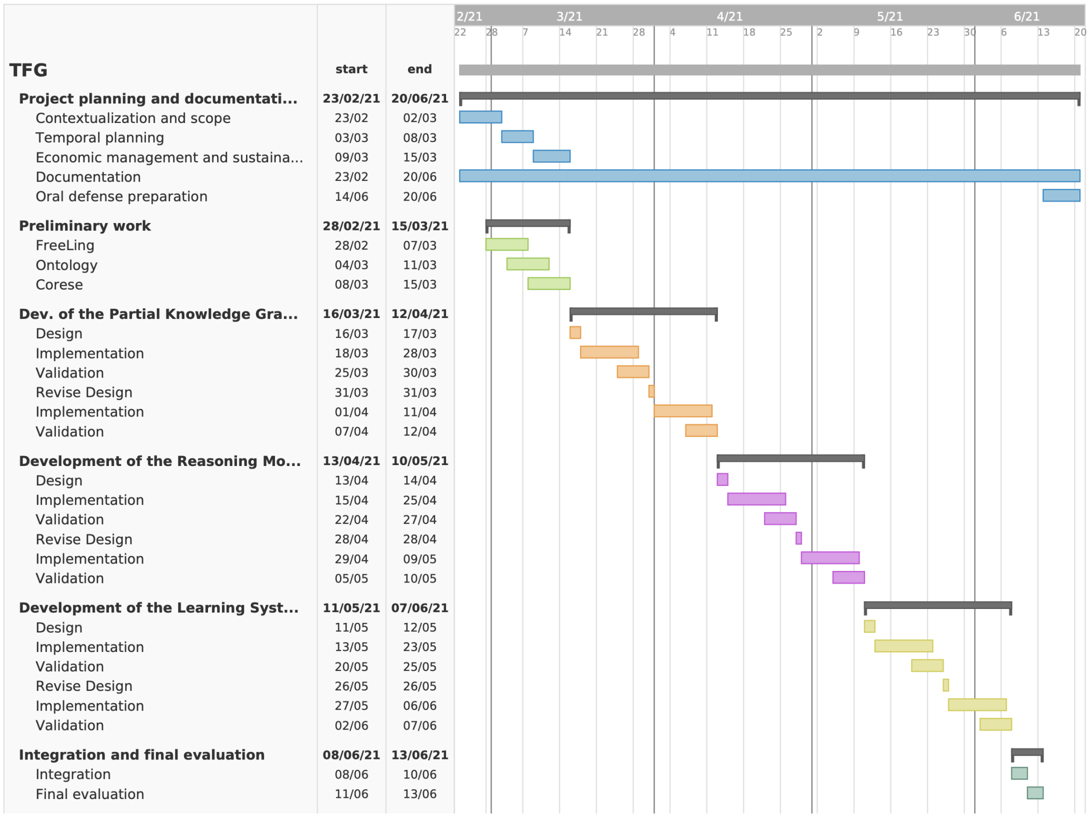
2.3 Initial Gantt

In the Figure 2-1, the Gantt chart is represented showing how the tasks and subtasks are distributed among the time frame and their required dependencies.

Meetings with the director will be scheduled in function of the development of the project. There will be the regular ones every two weeks, but there is the possibility of some extra ones to answer quick questions as they may appear.

2.4 Risk management

Some risks may appear through the course of the project and deviate its progress. Hence, in this section we evaluate the potential risks, how they can affect the project and how can be avoided or managed with the introduction of some potential solutions.



TFG	start	end
Project planning and documentati...	**23/02/21**	**20/06/21**
Contextualization and scope	23/02	02/03
Temporal planning	03/03	08/03
Economic management and sustaina...	09/03	15/03
Documentation	23/02	20/06
Oral defense preparation	14/06	20/06
Preliminary work	**28/02/21**	**15/03/21**
FreeLing	28/02	07/03
Ontology	04/03	11/03
Corese	08/03	15/03
Dev. of the Partial Knowledge Gra...	**16/03/21**	**12/04/21**
Design	16/03	17/03
Implementation	18/03	28/03
Validation	25/03	30/03
Revise Design	31/03	31/03
Implementation	01/04	11/04
Validation	07/04	12/04
Development of the Reasoning Mo...	**13/04/21**	**10/05/21**
Design	13/04	14/04
Implementation	15/04	25/04
Validation	22/04	27/04
Revise Design	28/04	28/04
Implementation	29/04	09/05
Validation	05/05	10/05
Development of the Learning Syst...	**11/05/21**	**07/06/21**
Design	11/05	12/05
Implementation	13/05	23/05
Validation	20/05	25/05
Revise Design	26/05	26/05
Implementation	27/05	06/06
Validation	02/06	07/06
Integration and final evaluation	**08/06/21**	**13/06/21**
Integration	08/06	10/06
Final evaluation	11/06	13/06

Figure 2-1: Gantt chart, made using TeamGantt

2.4.1 Inability to adapt existing components and tools

Given the intention to modify established tools and platforms, there is the possibility that the first plan to integrate directly the tools as they are and take the fastest route could not be finally implemented, implying a delay to find other options and re-doing some of the work.

- Impact: High
- Proposed solution: Dedicate hours to study the architecture of each part involved in the project in order to be more confident in what is doable. This can be taken care of in the preliminary work reserved hours with the purpose of inspecting the ontology structure, understanding the AI4EU's knowledge graph code or trying

the capabilities of the FreeLing's tool.

2.4.2 Inexperience in the field

The issue of getting involved in a new project is to not be properly familiarized with the concepts from the beginning, and required to start doing large tasks right away.

- Impact: Low
- Proposed solution: Beginning from doing the necessary preliminary research to get ready to approach the topic, while also adapting to the methodology of dividing the project in lesser dependent tasks to make the development more agile.

2.4.3 Ongoing platform

As AI4EU is a live project, changes to the ontology or other components might appear, so a strategy to combat and reduce the possible effects is going to be followed.

- Impact: Medium
- Proposed solution: Make the development easy to adapt to changes with a modular design in mind.

2.4.4 External factors

It is a reality that the entirety of the project will be done from home and even though we are almost used to work that way, at times it can be limiting.

- Impact: Low
- Proposed solution: Using tools to keep the communication with the director to ease the effects of the restrictions, and be ready to work from home with the proper equipment while also maintaining a good working environment and schedule.

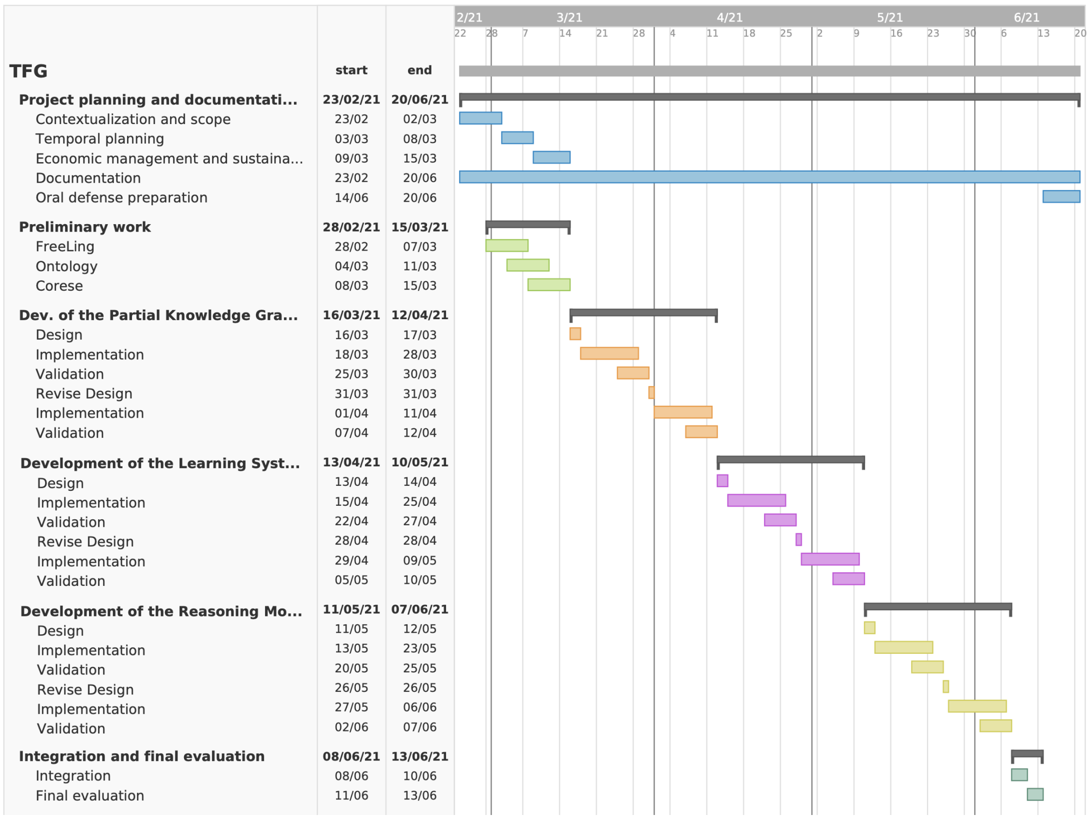
2.5 Final work plan

During the execution of the project there have been minimal changes to the original plan. Also, there have been no modifications to the objectives.

Implementation changes are something that any development project accounts for, as the planned design iterations and exploratory phases allowed for the time to choose ahead of time which direction take. As discussed in sections §3.1 for the changes to planned settings and the concretization of previously bare concepts, there has been decisions that were consensual with third parties, others that felt intuitive but many others required more deliberation time.

The minor changes to the temporal planning seem so natural in execution that is rare we didn't plan it that way from the beginning. Consisting of switching in place the development of the reasoning module for the learning system in the planning order, the reschedule didn't cause any delay. The swap rather has been beneficial because it improved development agility as it enabled the creation of an early functional prototype, facilitating the testing of the project as a whole. As for the timing of the tasks, the scheme was followed almost exactly and involuntarily, even sometimes begin starting the design of modules ahead of schedule. In the figure 2-2 below there's shown the updated Gantt chart including the described change.

As there has been minimal changes to this initial plan, and the refined context has not introduced any additional costs, the global costs stay the same and remain as calculated.



TFG	start	end
Project planning and documentati...	**23/02/21**	**20/06/21**
Contextualization and scope	23/02	02/03
Temporal planning	03/03	08/03
Economic management and sustaina...	09/03	15/03
Documentation	23/02	20/06
Oral defense preparation	14/06	20/06
Preliminary work	**28/02/21**	**15/03/21**
FreeLing	28/02	07/03
Ontology	04/03	11/03
Corese	08/03	15/03
Dev. of the Partial Knowledge Gra...	**16/03/21**	**12/04/21**
Design	16/03	17/03
Implementation	18/03	28/03
Validation	25/03	30/03
Revise Design	31/03	31/03
Implementation	01/04	11/04
Validation	07/04	12/04
Development of the Learning Syst...	**13/04/21**	**10/05/21**
Design	13/04	14/04
Implementation	15/04	25/04
Validation	22/04	27/04
Revise Design	28/04	28/04
Implementation	29/04	09/05
Validation	05/05	10/05
Development of the Reasoning Mo...	**11/05/21**	**07/06/21**
Design	11/05	12/05
Implementation	13/05	23/05
Validation	20/05	25/05
Revise Design	26/05	26/05
Implementation	27/05	06/06
Validation	02/06	07/06
Integration and final evaluation	**08/06/21**	**13/06/21**
Integration	08/06	10/06
Final evaluation	11/06	13/06

Figure 2-2: Updated Gantt chart, made using TeamGantt

Chapter 3

Design process

This chapter describes the architectural design of the proposed system. As the final design has departed from the original, planned design, first we show the early design to then discuss how early prototypes forced us to reconsider the role of some components and even the addition of new components.

3.1 Early Design and changes

3.1.1 The NLP engine

The development of the project has undergone some design changes, some significant but already foreseen and some minor that were not planned but that bring an overall improvement. The former is the case of the FreeLing integration. We expected to spend the first weeks trying to feed the ontology information directly into it and we ended up customizing its engine to be domain-specific. The results were not the ones we needed, so after a meeting with the tool's developer (Dr. Lluís Padró) and with his approval, the decision was made to scrap that solution to make use of the output data and build a separate processing module in conjunction with the program's built-in options, such as the dependency and semantic data.

The case for the proposed inclusion of a separated module to assist on NLP processing, ultimately required to inspect the FreeLing raw output and pre-process it to extract valuable knowledge for the other stages. PugiXML [17] has been the library used to implement the algorithms that collect and navigate the dependency and semantic data from the FreeLing XML output. For the previously mentioned ontology integration, Raptor [18] (an RDF parser for triple stores) has been used to explore the possibility of such unification, while preparing the first prototype with a built-in ontology in order to test out the text knowledge extraction to improve it.

An extension of the concept semantic model that was based on the CSO [8], which was already partly used originally, has been implemented as an alternative to the inclusion of the concepts into the NLP engine. This change means that the relations of terms and resources have to be specified somewhere in a new concept model, and such modifications have been added in order to maintain compatibility and create a separate module. Consequently, getting this expansion of the model, meant that a specific part knowledge graph had the one and only function to support the new resource classification reasoner. The change had to be supported by an existing database of terms, which conveniently the CSO has been selected as, after being analyzed in order to find a scheme that fits our problem, which can be adapted to complement the planned inference module.

3.1.2 The Knowledge graph

During the exploratory phase of the project and in the first weeks, we realized that the re-utilization of the prototype knowledge graph manager (Corese [16]) that was developed for the AI4EU project was not an option due to the demo nature of the public implementation. While maintaining compatibility with the existing platform via the RDF compatibility, the project scope had to be extended in two ways: not relying on the website visual representation and no performing SPARQL queries directly to the website.

The search for a graph managing system that suited our needs for replacing the previous method implied that many choices were in consideration during the election process. Apache Marmotta [19] was one of the first but the lack of current support and ambition were deciding factors compared to the final choice. One strong contender was Grakn [20], similar in concept and features to Neo4j, but with the big added benefit of a better-developed inference engine based on rules, which was the main attraction because of the possibilities it would bring. The decision was to settle with Neo4j [21] due to the graph visualization it offers and the compatibility with the back-end and ontologies used, even though it requires the exploration of plugins to reach the capabilities of semantic technologies and inference and reasoning engines.

Therefore, in our design the knowledge graph manager system will be backed by Neo4j [21] databases, which offer full compatibility with the RDF model standard. Using the Cypher [22] query language to store and retrieve data from the graph database, both the AI4EU categorical graph and the CSO [8] ontological information were imported and set up to support the implementation of an inference engine. The introduction of the concept of a resource node, and the new relationships that mirror their importance towards a set of topics to form an accurate representation, were required to reach the reasoning methods that we were looking for, even though it sets itself apart from the core concept scheme the knowledge graph philosophy of growing naturally has been taken into account. To administrate the Neo4j graph, the implementation of a Python driver, as a knowledge graph manager, had the utmost importance to both show a rich visualization (for explainability in the suggestion choices) and to make the flow of data between modules seamless.

For the visualization and interaction with the user, a simple HTML interface was already the intention but the addition of these new requirements to the module needed a back-end system to support them. The interface has to manage the users' queries and underpin the graph visuals and communication, and that made us consider the use of a scripting language such as JavaScript. Running as the server-side module, NodeJS [23] has all the benefits in need to reach a dynamic content web page and ExpressJS [24] the package facilitated the back-and-forth communication between the back-end and front-end. Additional frameworks such as the JQuery [25], enabled the HTML front-end to inspect itself and create different HTML events. The back-end, apart from ruling the HTML, is in charge of calling and intercommunicating with the rest of the modules, which are driving the NLP and graph.

As mentioned earlier, reaching the AI4EU's own full knowledge graph was a no option, thus a minor goal was to try to automate the resource descriptions' gathering, but the required sign-in proved to be a major obstacle as the intents with CLI tools like curl and with Oauth [26] methods failed. Coincidentally, as stated at the risks section, the web page has changed and with the update, many resources were removed (fortunately a backup was made beforehand) and now no longer a sign-in is required, meaning that a possible automated process would be feasible, but it is no longer needed.

3.2 Early Prototyping

The first prototyping phase started with the incorporation of the AI4EU ontology into the FreeLing software, intending to substitute one of the three included ones (WordNet [27], SUMO, and OpenCYC) to add upon the synsets [28] and the dictionary files for terminology. This method would have allowed for rapid development since it only required inserting that specific domain, but in the long term, it would have had a lot less flexibility compared to the final solution. So we switched to a different solution where a pre-processing module was implemented in C++, and thanks to it we were able to reach reach a first working prototype (see §3.1). Parsers and XPath [29] were used to fuse NLP and ontological data and match them via string distance functions such as Levenshtein. As the standard matching was working, term databases had to be considered to add a base to build upon, starting from the ones mentioned in §1.1.2, the testing started with the CSO using broader and narrower topics as simple recommendations.

Interfacing with the user was the second problem approached, and as soon as a graph manager was decided, the graph visuals were incorporated in the same phase. As stated in §3.1, a back-end was key to the increasing complexity of different programs and drivers which had to talk to each other and finally interact with the interface, so a lot of thought was put into building that foundation from the start.

The prototyping phase of the reasoning module is discussed in the §4.3 section in detail.

3.3 Final Design

In this section we define the architecture of the proposed resource classifier program, to accomplish the learning based on user interaction.

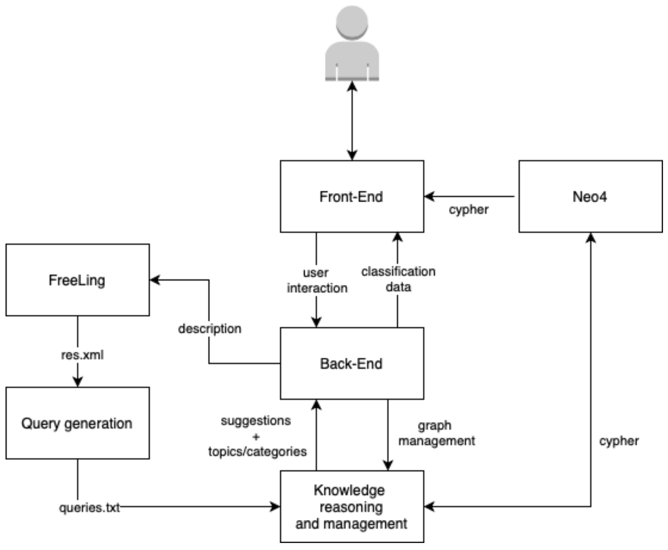


Figure 3-1: Design of the proposed architecture

3.3.1 NLP Pre-processing (FreeLing + Query generation)

The main goal in using the FreeLing software is to analyze the input text and extract a Semantic Graph that suits our needs of representing the valuable text information, in order to carry on the further analysis. The settled strategy, as stated in §3.1.1, has differed from the original of cross-referencing the ontology and the provided semantic graph, into a custom-built graph that takes into account additional NLP data.

Dependency word information is key to construct a reasonable matching method between terms and the ontological data, so we make use of the provided tree to build a partial graph (see §4.2.1) also including some semantic data. With this graph processed and complete, we are able to make the query the main knowledge graph.

3.3.2 Knowledge Graph Management (Knowledge Reasoner & Manager + Neo4j)

The chosen Knowledge Graph database has been Neo4j, which is a graph database platform that has checked all the requirements, as explained in §3.1. Feeding into the ontological data from both the AI4EU categorical and CSO terminology, following the planned concept model and building upon it with new user interaction obtained intelligence.

As for the instructed queries received from the NLP module, they are matched and/or used for ontological reasoning. Using a series of inference methods, §4.3, new suggested data is recommended to the user backed with the graph reasoning. The other way is allowed too, enabling the user to adjust and insert new information to improve the graph representation.

3.3.3 Visualization (Front-end + Back-end)

Communicating with the knowledge graph manager, a visualization service has been developed to interact with the user and let it explore and add AI resources.

It consists of a back-end (that generates most of the content of each shown page) and the front-end (on the user's browser, able to dynamically modify some elements directly on the page wothout re-loading). The front-end includes a rich visualization of the graph (provided by the Neovis [30] library) and text fields with several dynamic features such as selectable text or term highlighting, among other features, build upon an HTML plus NodeJS [23] structure.

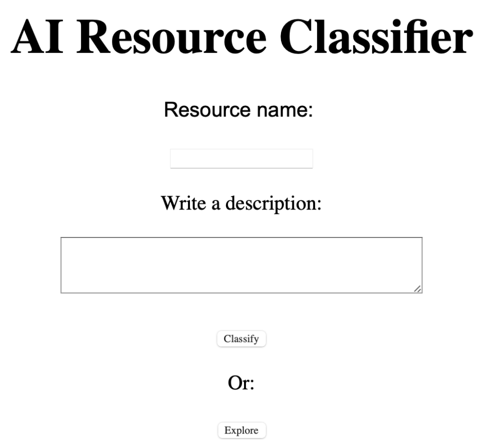


Figure 3-2: Introductory view of the classifier

Chapter 4

Implementation

This chapter describes all the different elements that have been developed in this project. We start by describing the pre-existing tools and components that we use as basis (highlighted in gray in figure 4-1), and then we describe all the elements developed during the lifetime of this project (white in figure 4-1).

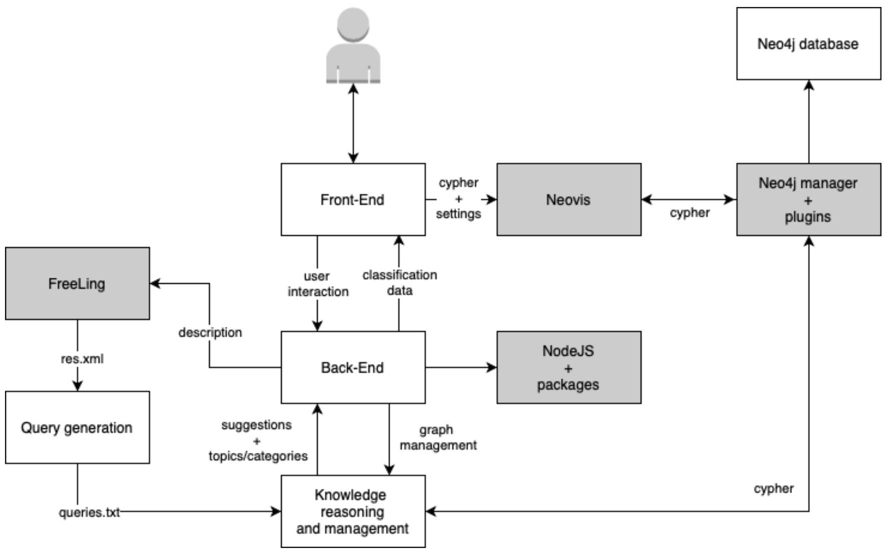


Figure 4-1: Architecture of the implemented system.

4.1 Starting point: existing tools and components

4.1.1 FreeLing

Using the analyzer program from FreeLing to process corpora required an analysis of what we wanted it to extract from the text and how we can get it, with the former already being explored in the starting design phase §3.1.1. The engine provides a series of analysis options that have been used as a base such as the number, date/time or quantities/ratios/percentages recognition to both identify and separate them from the important terminology we are searching for. In order to find the terms named entity recognition has been enabled and used alongside multi-word detection to match them with the said entities. These parameters assist the NLP engine to provide refined semantic and dependency analysis levels.

On the semantic level, WordNet [27] sense annotation has been taken advantage of to utilize the big database of general meanings FreeLing has to help build our semantic graph. Moreover, sense disambiguation is of utmost importance for an accurate domain selection of senses, even if for very specific cases may lack sensibility, in the majority of cases is worth it meaning that the sense is identified correctly. The disambiguation method used is UKB [31], which is a graph-based word sense disambiguation with lexical similarity using pre-existing knowledge database like the previously mentioned WordNet [27].

On the dependency level, FreeLing includes several dependency parsers which each one build using a different technology. The first one is rule-based which is the faster option but not as accurate, the second, statistic-based, is slower and aided with machine learning. The chosen one, which is the most accurate and is also faster than the statistical one, is a neural-network-based dependency parser using LSTM [32] techniques.

On the performance side of the tool, we do not expect instant text analysis results because it is an intensive review being made which means that it takes in the order of tens of seconds. There is the possibility of threading the work here using the threaded analyzer tool, but in its current state it does not support all the analysis options in need, additionally, it would need a lot of development of our part to make it work currently.

4.1.2 Neo4j

As for the generated graph utilities used in Neo4j, a couple of plugins have been incorporated to the graph manager such as Neosemantics [33], APOC [34], and Graph Data Science Library [35], with the latter following the accompanying book [36] and documentation to learn its possibilities as seen in the §4.3 section.

Neosemantics, has mainly been of utility to offer the RDF and concept semantic model compatibility with the ability of dynamically exporting and importing ontologies. The added benefit of Neosemantics is that it offers the Neo4j base inference semantic-based engine, which is being used in the reasoning module §4.3. Support graph management tools such as creating an inverse of relationships is done via the APOC [34] (Awesome Procedures On Cypher) add-on.

4.1.3 NodeJS

The interface side of the program had the need of intercommunication between the back end and the front end with the ability of interactive modules that react to user input.

First of all, the Handlebars [37] NodeJS module is in charge of managing the different interface views, from the main input screen to the different forms of the result view. Each section, requires specific data being carried from the back end to the front end, and the Handlebars manager is able to pass it to the HTML template, these includes all the knowledge graph manager topic and categorical suggestions and also the resource description and title among others.

The user's interaction with the recognized text is done via the selectable description, which also includes colored identified topics, seen on §5. In order to be able to click single words or select a compound term, the Jquery [25] library introduces the `click` and `mouseup` functions coupled with the ability to get the proper window and selection information from the interaction calls. Furthermore, Jquery enables the creation of dynamic HTML posts used in the suggestions buttons or any knowledge graph manager related task from the front end.

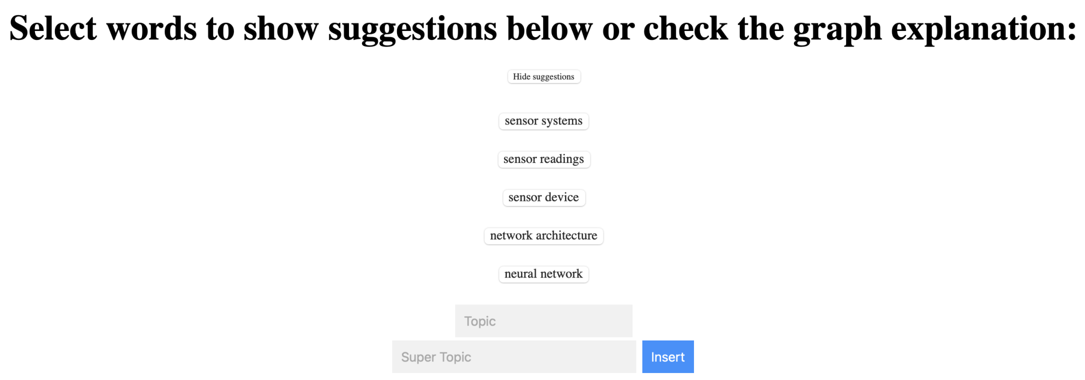


Figure 4-2: Dynamic suggestions plus new topic addition form (included with auto-complete function)

4.2 Developed components

4.2.1 Dependency processing (Query Generation module)

One of the elements we had to develop was the one right after the FreeLing analysis that could process the dependencies provided by the NLP output. After some tests in the early prototyping phase, the best method consisted of extracting multiple sub-graphs for each couple of tokens which, alongside semantic data, will be the source of graph matching and inference. Using the dependency part of the XML output from Freeling and the PugiXML [17] library §3.1, a series of nodes and combinations have been selected for a multi-word size of at least 2, while also omitting stopwords from the NLTK [38] list to get a cleaner result.

The nodes chosen are based on combinations of parent and sibling dependent nodes from the iterated source, and they are mainly explored from a bottom-up perspective. The first nodes added are combinations of the actual term plus the parent word, and also with the sibling's words, in order to collect the two main topics like on the first output below. Supported by the results obtained in the FreeLing dependency tree, and as seen on the second output below, tests showed that there was a need of adding more relationships to the combinations (such as a couple of parent sibling words) to get more accurate and broader results. Additionally, order has been taken into account with the two different possible arrangements for each case while also including the semantic data in each term. This method gets us a precise representation of the description and provides a complete list of queries (section §F) to use with the graph.

```
 <depnode token="t1.24" function="ADV" word="with" >
   <depnode token="t1.25" function="PMOD" word="machine" />
   <depnode token="t1.26" function="PMOD" word="learning" >
     <depnode token="t1.27" function="COORD" word="and" >
       <depnode token="t1.28" function="CONJ" word="artificial" >
         <depnode token="t1.29" function="COORD" word="intelligent" />
         <depnode token="t1.30" function="COORD" word="in_order_to" />
       </depnode>
```

```
<depnode token="t6.9" function="COORD" word="and" >
        <depnode token="t6.10" function="CONJ" word="develop" >
          <depnode token="t6.11" function="OBJ" word="machine" />
          <depnode token="t6.13" function="OBJ" word="algorithms" >
            <depnode token="t6.12" function="NMOD" word="learning" />
            <depnode token="t6.14" function="NMOD" word="to" >
              <depnode token="t6.15" function="IM" word="recognize" >
                <depnode token="t6.17" function="OBJ" word="behaviors" >
                  <depnode token="t6.16" function="NMOD" word="the" />
```

4.2.2 Expansion of the concept model (Neo4j database)

The expansion of the original concept model has kept in mind two priorities, the support of the new reasoning module as a separated entity and the coherence with existing sections of the AI4EU's ontology.

As stated in §4.1.2, two ontologies in Turtle syntax as an N-Triple file format, the CSO and AI4EU, were imported into the Neo4j graph database. The AI4EU's ontology is the categories-only version, which is the single provided file with included instances and is also a useful starting point which to expand upon. To interconnect these two types of ontological data, topics, and categories, there is the introduction of a new node called "AI RESOURCE" different from the included in the original model as this new version is the analyzed version which contains the dynamic topic and categorical relationships which supports the reasoning module. The classification node bridges to both AI4EU and CSO with two new relationships, "HAS TOPIC" and "HAS CATEGORY", declaring to which topics or categories might contain and form the learning knowledge graph.

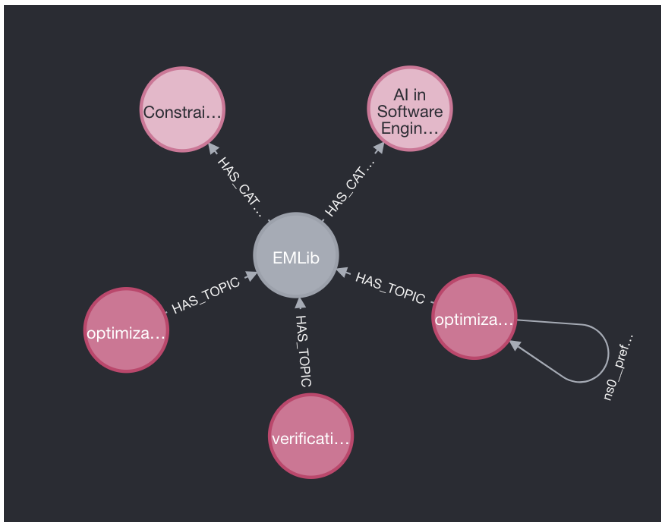


Figure 4-3: AI Resource node related to both topics and categories

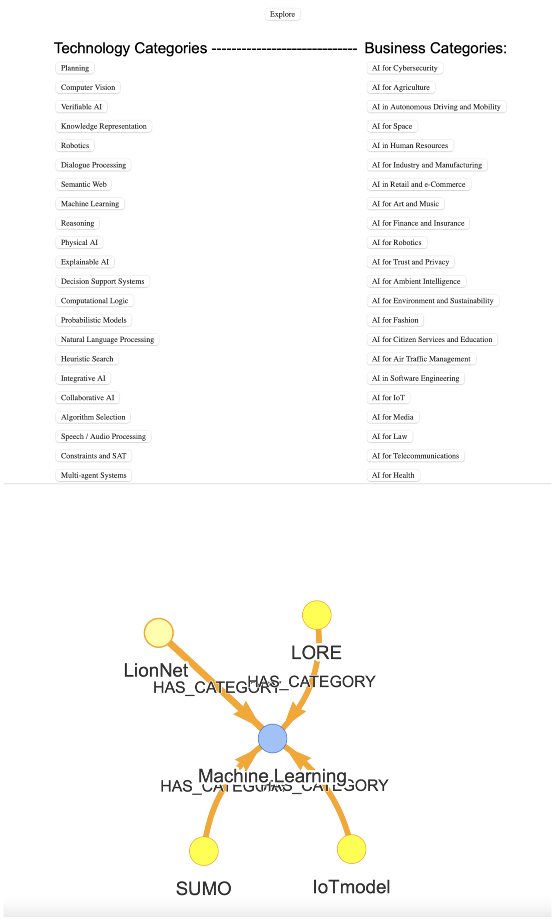
By not using the original, rigid, AI4EU ontology and creating our own extension enables the graph to grow independently and provide an assistant role to the main AI4EU ontology. In future work it may be explored a supervised mechanism to introduce the new learned nodes in the main AI4EU ontology, but in this thesis we focus on the learning process for the entities, not in the update of the static project's ontology, as the process to update the AU4EU ontology is is actually an internal topic of discussion inside the project partners.

4.2.3 Graph views (Front-end + Back-end modules)

The graph visualization system has two different views: explore and result

Explore view

On the initial step of the resource classifier program, aside from entering a new resource, there is the option of exploring existing resources by category which enables the user to get a sense of the graph's categorical information.


Explore

Technology Categories ---------------------------- Business Categories:

Technology Categories:
- Planning
- Computer Vision
- Verifiable AI
- Knowledge Representation
- Robotics
- Dialogue Processing
- Semantic Web
- Machine Learning
- Reasoning
- Physical AI
- Explainable AI
- Decision Support Systems
- Computational Logic
- Probabilistic Models
- Natural Language Processing
- Heuristic Search
- Integrative AI
- Collaborative AI
- Algorithm Selection
- Speech / Audio Processing
- Constraints and SAT
- Multi-agent Systems

Business Categories:
- AI for Cybersecurity
- AI for Agriculture
- AI in Autonomous Driving and Mobility
- AI for Space
- AI in Human Resources
- AI for Industry and Manufacturing
- AI in Retail and e-Commerce
- AI for Art and Music
- AI for Finance and Insurance
- AI for Robotics
- AI for Trust and Privacy
- AI for Ambient Intelligence
- AI for Environment and Sustainability
- AI for Fashion
- AI for Citizen Services and Education
- AI for Air Traffic Management
- AI in Software Engineering
- AI for IoT
- AI for Media
- AI for Law
- AI for Telecommunications
- AI for Health

Figure 4-4: Explore view category filters

Result view

Offered when a resource is classified and waiting user input, the result graph view shows an overview of the recognized terms and their similarities with any connections between them. As seen in 4-5, the main resource is marked in red, and in yellow it is shown the user inputted topics either from the actual classification or previous ones.

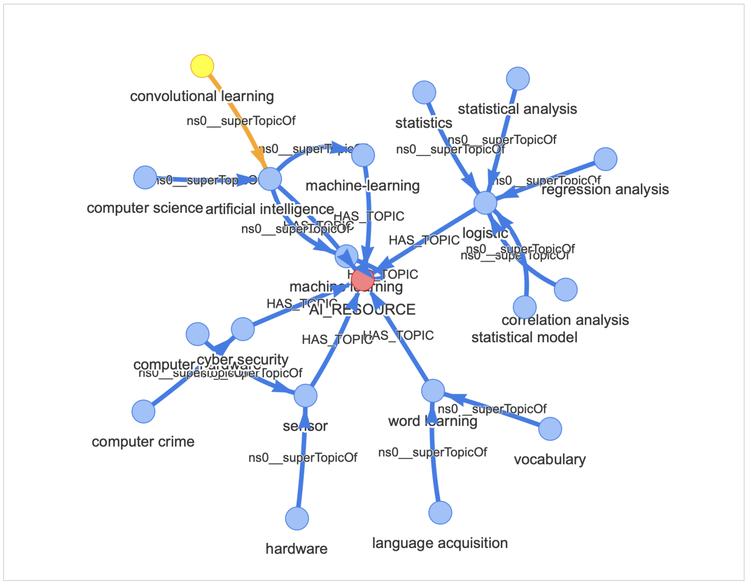


Figure 4-5: Neovis graph with new topic

4.3 Implemented algorithms (Knowledge Reasoning & Management module)

The objectives of the algorithms made for the reasoning module are to learn from new experiences and relate resources to make suggestions. A series of queries and procedures have been designed to take advantage within the Neo4j capabilities of the built knowledge graph. These techniques are categorized into two groups: semantic and similarity, additionally, each one is coupled with an extra method in order to narrow them down.

4.3.1 First semantic method

The first inference section of the reasoning module that was designed, is the conveniently named "first semantic" method. Trying to make use of all the sub and super relationships between topics of the CSO ontology and the Neo4j built-in nodesInCategory inference call. Consisting of a search from a top level category, it finds all its transitive subcategories, and finally then returns nodes attached to any of those categories. This categorical query is feed with "in category" relationships as the ontology "prefrentielEquivalent" and with "sub category" ones with "superTopicOf" (which for this operation has been inverted temporarily to a "subTopicOf", in order to align with the inference function requirements).

Resulting in a broad selection of clusters of sub-topics linked to the main request in order to be refined in a following algorithm. The purpose of this method is to map a broad definition of what the categorical shape of the resource topics might look like.

CYPHER QUERY:

```
MATCH (c:ns0__Topic {rdfs__label: topic_name})
    CALL n10s.inference.nodesInCategory(c, {
      inCatRel: 'ns0__preferentialEquivalent',
      subCatRel: 'ns0__superTopicOf'
    }) "
YIELD node
WITH node.rdfs__label as nodes
RETURN nodes
```

4.3.2 Second semantic method

With a base for the semantic inference to suggest from what is already recognized, the second semantic intention is to build upon what may be left unrecognized. Improving the term identification with semantic data from the ontology with similar suggestions of the type "Would you mean x?". As an example in the figure 4-6, supposing a text about the internet that also addresses topics about streaming throughout it, the module would reason that you are meaning something about "video streaming" or "streaming media".

The Neo4j query used to get such results is based on paths from the inputted broad term, "internet" in this case, to any combination of "superTopicOf" relationships until any of them or their "relatedEquivalent" contains the narrow term, "streaming". The result may or not be bounded enough depending on the graph relationships and density, therefore on a following algorithm we will explain how we solve it.

Moreover, the use of paths in Cypher queries is a technique used broadly in the whole reasoning module, moreover, it was a great surprise at how intuitive and easy to come up with knowledge extraction methods.

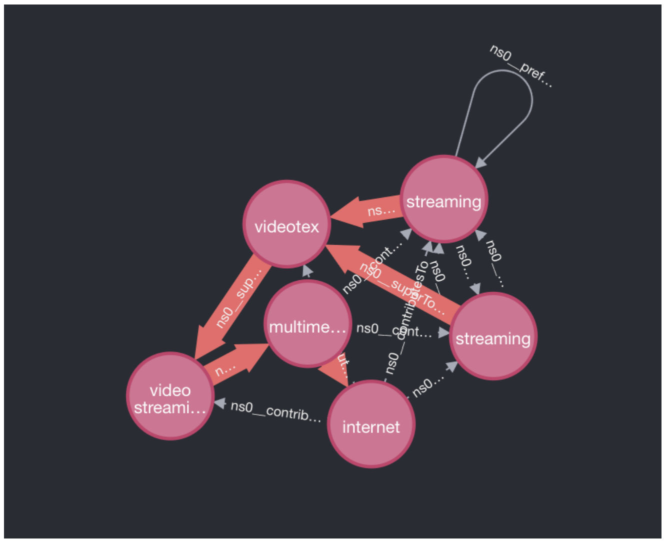


Figure 4-6: Second semantic output Neo4j graph

CYPHER QUERY:

```
MATCH path = (c:ns0__Topic)-[:ns0__relatedEquivalent]->(category)
-[:ns0__superTopicOf*]->(:ns0__Topic {rdfs__label: broad_term})
WHERE c.rdfs__label contains concrete_term
RETURN c.rdfs__label as name
```

4.3.3 First similarity method

Getting on to the first query that takes advantage of similarity aspects, it has to be noted that to appreciate its results it needs other resources topic classifications. The motivation of this method and the "second similarity" is to nurture from learned users interactions to guess topic recommendations based on a series of factors.

First of all, we are searching for a collection of resources that include similar topics from ours. With each of these alike resources of varying degrees, we extract these new foreign topics. Normally the result is very much verbose but as we will show later with the added refinement it makes all the sense to include it.

The double path implemented on the Cypher [22] query goes from the actual resource to the similar resource traversing from common topics and then matching the related resources novel topics.

CYPHER QUERY:

```
MATCH (c:AI_RESOURCE) WHERE id(c) = $res_id
WITH (c) MATCH path = (c)<-[:HAS_TOPIC]-(topic),
    otherPath = (topic)-[:HAS_TOPIC]->(res)
MATCH (res)<-[:HAS_TOPIC]-(newtopic)
WHERE newtopic <> topic
RETURN newtopic.rdfs__label AS name
```

4.3.4 Second similarity method

The "second similarity", very akin to the first, with the difference of taking advantage of categorical types, as the previous took the path downwards with "HAS TOPIC" relationships, in this case, it goes upwards into "HAS CATEGORY" territory. Therefore, the intention is to collect the topics that appear on resources classified within the same category.

CYPHER QUERY:

```
MATCH (c:ns0__Topic {rdfs__label: $topic_name}),
    path = (c)-[:HAS_TOPIC]->(res)-[:HAS_CATEGORY]->(cat),
    otherPath = (other)-[:HAS_CATEGORY]->(cat)
return path, otherPath
```

4.3.5 Centrality methods

For this and the next refinements, Graph Data Science plugin methods have been selected for each one. Applied to the first semantic, the centrality algorithm refinement aims to reduce the multiple clusters, generated from the inference of subcategories of topics, into a couple of defining nodes.

Centrality algorithms [36] are used to determine the importance of distinct nodes in a network form such as a graph. Among a multiple set of choices, both the PageRank and Betweenness have been implemented and its values for each node calculated on the knowledge graph. Ultimately, the tests done in §5 have utilized PageRank values.

The PageRank [36] algorithm measures the importance of each node within the graph, based on the number of incoming relationships and the importance of the corresponding source nodes. The underlying assumption is that a page is only as important as the pages that link to it. PageRank was introduced in an original Google paper as a function that solves the following equation:

$$PR(A) = (1-d) + d(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)})$$

where,

- we assume that a page A has pages $T_1 ... T_n$ which point to it
- d is a damping factor which can be set between 0 and 1
- C(A) is defined as the number of links going out of page A

The other option, Betweenness [36], is another centrality algorithm that consists of detecting the amount of influence a node has over the flow of data within the graph.

Actually, it is commonly used to find nodes that serve as a bridge from one part of a graph to another. The algorithm calculates unweighted shortest paths between all pairs of nodes in a graph, with each node receiving a score, based on the number of shortest paths that pass through that node. Thus, the nodes that more frequently lie on the shortest paths between other nodes will have higher betweenness centrality scores.

4.3.6 Community detection method

Applied to the second semantic method, the use of communities restrict the expansion of suggestions based on the combination of topics. It is only useful if the recommended blend of topics stays within a certain community otherwise unexpected and not accurate terms may appear. In order to face this phenomenon, community detection algorithms serve to evaluate how groups of nodes are clustered or partitioned.

The Louvain [36] method has been chosen as an algorithm to detect communities in our knowledge graph. It maximizes a modularity score for each community, where the modularity quantifies the quality of an assignment of nodes to communities. This means evaluating how much more densely connected the nodes within a community are, compared to how connected they would be in a random scenario. This algorithm is a hierarchical clustering type of algorithm, which recursively merges communities into a single node and executes the modularity clustering on condensed graphs.

4.3.7 Link prediction method

Applied to the first similarity method, it tells apart which topics pertain to a closer representation of our resource scenario and filters out many of suggestions that do not have that much in common. We can take advantage of how close these guessed nodes are to our topics and try to get the ones that align with the shape made from our resources found matches.

Link prediction techniques help determine the closeness of a pair of nodes, as their computed scores can then be used to predict new relationships between them. The algorithm used, Adamic Adar [36], was introduced in 2003 by Lada Adamic and Eytan

Adar to predict links in a social network, and it is computed using the following formula:

$$A(x, y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

where N(u) is the set of nodes adjacent to u, and the output score values range from 0 indicating that two nodes are not close enough to higher values showing how nodes become closer.

4.3.8 Node similarity method

Applied to the second similarity method, it solves the need to know which of the supposed similar topics within common categories are the most alike and thus on-point recommendations. Applying a similarity algorithm allows us to compute the semblance of pairs of nodes using different vector-based metrics.

The node similarity method compares a set of nodes based on the nodes they are connected to, therefore two nodes are considered similar if they share many of the same neighbors. Then, the Jaccard [36] metric is used to compute the pair-wise similarities and obtain the corresponding score, which is computed using the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

where the input of the function is a bipartite, connected graph containing two disjoint node sets, with each relationship starting from a node in the first node set and ending at a node in the second node set.

The goal is to compare each node that has outgoing relationships with each other such node, such that for every node n, we collect the outgoing neighborhood, N(n), as all nodes m to which there is a relationship from n. Then, for each pair n and m, the algorithm computes the similarity score for which the Jaccard function is applied on both N(n) and N(m).

Chapter 5

Testing

In this chapter some tests of the proposed system and the results obtained will be shown and discussed.

It is important to note here that testing a system able to learn at runtime is not easy, as you will never get the same results after repeating the same interactions with the system. For instance, if a user introduces an AI resource with a new deep learning technique not present in the knowledge graph, the system will not be able to suggest that technique to toe user. But after learning the concept, introducing the same or a similar AI resource will have a different behaviour (the system suggesting the new technique to the user).

So the main objectives in this chapter is to show not only how the system is able to tag resources with the pre-existing concepts, but the learning capability of the system to incorporate new concepts and use them in the next interactions.

5.1 Real-world data

In order to get real-world data from resources, we looked to AI4EU website's public data. A group of top 10 resources and other different resources has been formed to offer a representative picture of the multiple use cases. Ranging from categories related to media or traffic control to common ones as cybersecurity or computer vision, the system has to learn the resources relationships to be able to suggest topics accordingly, as it will be shown in the following section.

5.2 Result analysis

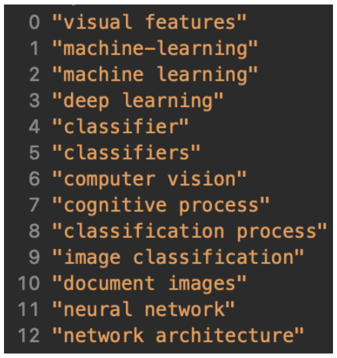
The testing methodology follows the stated group of resources with a selection of 10 different ones and then capturing the output results, such as matched topics and category suggestions. More importantly, the range of topic suggestions has been divided by the algorithm that has been used to find it for better comprehension of the outcome.

The full results that back the following sections can be found in the §G section. Also, both the used resources and the source code are linked on §A and included as additional material, with the resources having the original keywords and categories inputted by the original author as a reference.

5.2.1 Topic and category recognition

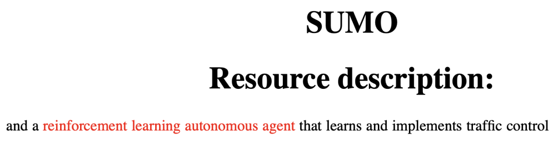
For the topic and category recognition, the matching level used for the Levenshtein Distance Algorithm was 0.9 which is very conservative but found accurate. The needed improvement is to be able to identify more topics than from what is initially established on the database. For this reason, the *add topic* function was added and works very well to assist the matching module in order to improve future classifications.

As for the state of the recognition topics, it is somewhat acceptable as seen below for an example resource, the document images classification.



```
 0 "visual features"
 1 "machine–learning"
 2 "machine learning"
 3 "deep learning"
 4 "classifier"
 5 "classifiers"
 6 "computer vision"
 7 "cognitive process"
 8 "classification process"
 9 "image classification"
10 "document images"
11 "neural network"
12 "network architecture"
```

Figure 5-1: Multiple topics being found on a document images classification resource



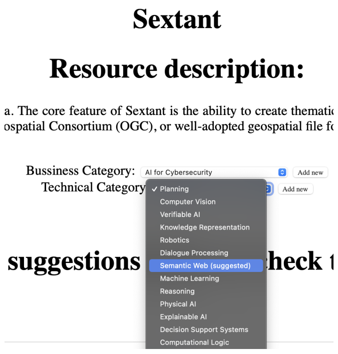
SUMO

Resource description:

and a reinforcement learning autonomous agent that learns and implements traffic control

Figure 5-2: Compound of terms being recognized as a topic

In the case of category matching, the technical ones are the most accurate of the two, being almost always spot on. In contrast, the business category suggestion was mostly empty because of the broad range of significance each one carries and we are not able to get useful recommendations most of the time with the base algorithm. Two examples of these occurrences are:



Sextant

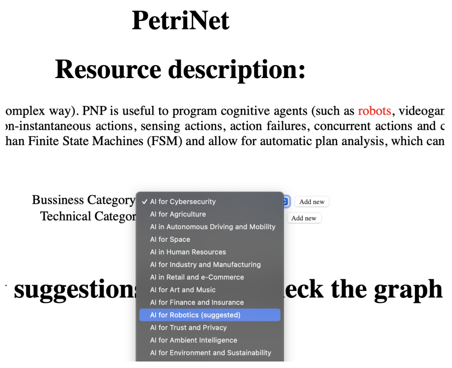
Resource description:

a. The core feature of Sextant is the ability to create themati
ospatial Consortium (OGC), or well-adopted geospatial file fo

Bussiness Category: AI for Cybersecurity
Technical Category: Semantic Web (suggested)

suggestions **heck t**

Figure 5-3: Business category suggestions


PetriNet

Resource description:

omplex way). PNP is useful to program cognitive agents (such as robots, videogar
n-instantaneous actions, sensing actions, action failures, concurrent actions and c
han Finite State Machines (FSM) and allow for automatic plan analysis, which can

Bussiness Category
Technical Categor

suggestion eck the graph

✓ AI for Cybersecurity
AI for Agriculture
AI in Autonomous Driving and Mobility
AI for Space
AI in Human Resources
AI for Industry and Manufacturing
AI in Retail and e-Commerce
AI for Art and Music
AI for Finance and Insurance
AI for Robotics (suggested)
AI for Trust and Privacy
AI for Ambient Intelligence
AI for Environment and Sustainability
Add new
Add new

Figure 5-4: Technical category suggestions

To fix category mismatches or unavailability of certain ones, two "Add new" buttons have been conveniently added to manually introduce new categories into the knowledge system, in order to better classify the said resources.

5.2.2 Suggestion evaluation

Starting with EMLlib as the first resource added, and as expected, only the two semantic offered new recommendations since the two similarity methods depend on other resource data. The first semantic offers a set of suggestions based on the sub categories for "optimization" and "verification":

```
"optimization": ["global optimal solutions", "discrete particle swarm optimization",
"assignment problems", "quadratic assignment problems",
"local optimal solution", "discrete particle swarm optimization algorithm",
"lyapunov methods", "assignment problem", "quadratic assignment problem",
"lyapunov method", "knapsack problems", "mutation operators",
"hybrid particle swarm optimization", "improved particle swarm optimization",
"valid inequality"]
"verification": ["model based testing", "run-time verification",
"model-based testing", "runtime verification",
"formal verifications", "formal verification"]
```

At first glance, in the case of `"optimization"`, only the first elements are the most accurate being the rest of simple guesses that may or not be useful, this effect is due to being ordered by centrality. In the `"verification"` case, almost all are good recommendations because there is not much variances as observed owing to not being a broad topic like the previous one. Unfortunately, the second semantic does not offer anything new based on the surrounding words, partly because it is a short description.

Learning new topics

The point that we are trying to get across is going to be shown more clearly as more resources get introduced in the system, so we fill out the system with the said resources and try to see the improvement made and from where they come. Both, the group of resources and every step relating to the progress by the addition of resources will be found on the projects folder under testing results.

Now with 8 more classifications added, we are going to test an IoT data analysis model related resource with supposedly links to previously added resources. This time, we get the four methods full of results and with three main topics identified: sensor, logistics, machine learning, and IoT. In this case, we explore continue exploring from the second semantic algorithm, paired with a longer resource description now finds out relations to "logistics" and "sensor". For "logistic" it finds related terms for surrounding words such as industry and enterprise based on semantics and bounded with the detected communities, but in the "sensor" case, the result does not get us anything beneficial as broad topics such as device or readings are returned and not the expected "motion sensor". This is due to not being on the terms database, so the easiest solution is to add it via the topic creation functionality. With this done, the second semantic finds connects motion and sensor and returns the expected recommendation and visualization.

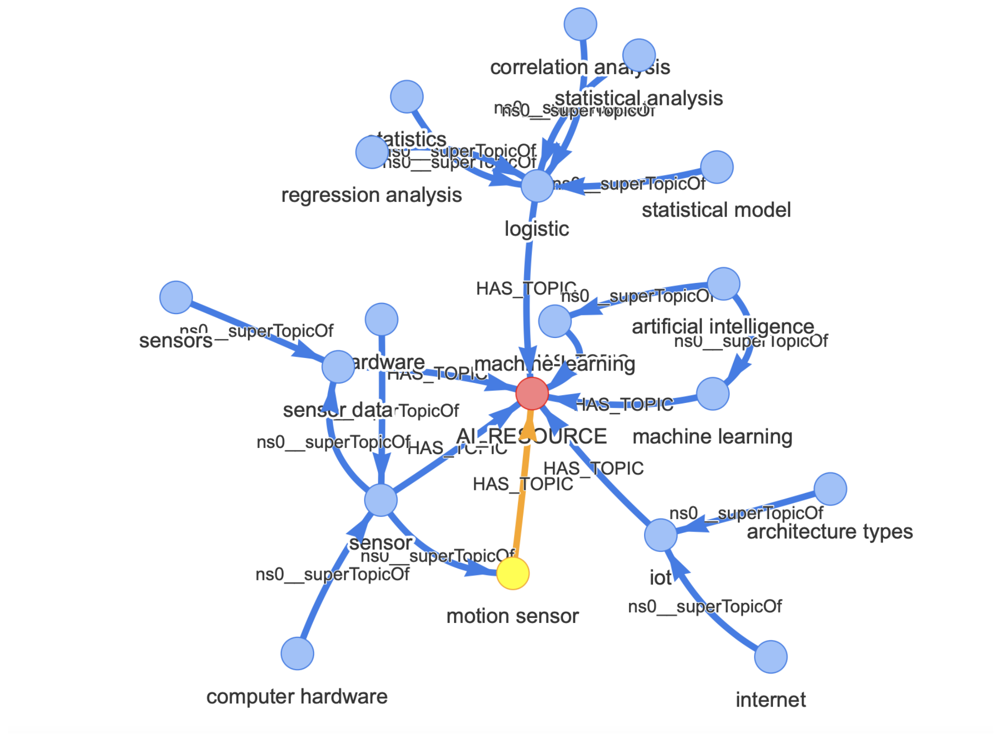


Figure 5-5: "`motion sensor`" recognized via the second semantic method

For the similarity algorithms, we need to take into account previously added resources because are the main source of learning. The first similarity which is topic-based, brings us interesting recommendations not provided from the previous methods, two examples among others are:

```
"machine learning": deep learning and neural network (from document
images classification), classifier (from LORE)
"sensor": computer vision (from smartROAD), network architecture (from
document images classification)
```

There is visible learning and order of affinity via the added link prediction of the added recommended topics, which some are apparent on various other resources as well.

In the case of the second similarity which is category-based, the suggestion is more precise as it only gives similar topics included among the same resource category

```
"machine learning": ["classifier", "neural networks"]}
```

If we were to look at this machine learning recommendation before the node similarity refinement, it would give us almost every topic recognized under almost all previous resources as many of them were tagged as Machine Learning under the technical category. In this case, the improvement is done through only selecting the suggestions that share many of the neighbors as our resource topics.

As a rundown of the inference capabilities and its results, it has been demonstrated that every method has its place and the added refinement contributes to the correct performance with reduced verbose recommendation data as possible.

Chapter 6

Conclusions and future work

The initially defined purpose of this project was to create a prototype of a knowledge graph-based learning system that could improve with user interaction. The final system developed in the project has not only accomplished this but it also completed all the planned objectives within the established technical context while also managing several design changes along the way.

Through the development of this project, it has been achieved a certain degree of learning capability with the design of the four methods within the reasoning module. As proved, each of these techniques show beneficial suggestions paired with their refined evaluation. Having piqued an interest in the kinds of graph algorithms explored, the design process of which has been kind of experimental, there is room for further improvements and tweaking, even though it is consistent with what we set out to make. The real asset we got from the project is what we learned from different competences (see Annex B) and how we managed to apply them to build a functional tool. Also, the work done to build a base to sustain and present both the graph and recommendation systems has resulted in an insightful interactive platform.

There are some future lines of work. The most obvious one is the integration of this prototype into the AI4EU web portal. During the lifetime of this project, AI4EU has re-developed a completely new portal using Drupal. Our choice of Neo4j database and components makes it easier such integration, but there are two areas that require some research: 1) the implementation of the dynamic features of the front-end in the new Drupal-generated webpages, and 2) the semantic alignment between the dynamic knowledge graph structure we propose in this document and the new AI4EU ontology that is being re-developed now.

But the results of this work may be applicable to other systems and domains. For instance, the development and use of knowledge graphs on Recommender Systems and information retrieval engines has a great prospect on its combination with graph neural networks to solve completion problems. Furthermore, one unreachable goal has been to improve the recognition one more level with the introduction of massive data from hundreds of resources at once. Intentionally (see Annex C), this prototype has been made available to anyone interested on integrating it with their classification system wanting to take advantage of Knowledge Graphs.

Appendix A

Source code

The project's source code has been included as additional material, or it can also be found on this public repository: https://github.com/arnau9marti/classifier Be sure to check out the readme text file for more information on how to set it up and how the modules are distributed among the code files.

Appendix B

Integration of knowledge (Technical competences)

Aspects of a lot of disciplines were used to develop what this project aims to solve. The most obvious are AI topics such as inference and reasoning or ontology knowledge, learnt useful approaches but I had to apply in a much different context with new languages and platforms and even diverse ontology formats. Referring to databases, an implicit previous experience of building queries in other much different situation have been helpful to the building of the knowledge graph system.

The usefulness of having to develop in any programming language needed for an specific part of the project, such a parser or a driver or even front end, is something that during the formation I have been gathering randomly, but more concretely from the programming classes. But what is the code's function without an underlying algorithm, learnt mostly in algorithms lectures, and being used in NLP processing to travel the dependency tree or extracting useful semantic data to build the multi-term queries.

Appendix C

Identification of laws and regulations

From the start of the project definition, the use of open source tools and from the UPC, such as FreeLing, was defined as the base to build upon it. The ability to inspect the code and even change how it manages certain features towards adapting it with our problem has been a focal point. Mainly, under various forms of GNU Public Licences and GPL-3, for the FreeLing and Neo4j software, Apache licence for certain parsing tools and even MIT licence XML possessing. Most of the licences mentioned offer the developer to deal with the software without restriction and without limitations to the rights to use, copy, modify, merge, publish, distribute it. The trade-off usually is the fact that it does not come with any warranty meaning that is distributed "as is", but that is not an impediment in our case as the software is constantly being tested an modified as needed.

In terms of management of the data being used and dealt with, in this case, we are covered under European jurisdiction meaning that General Data Protection Regulation 2016/679 (GDPR) has to be followed in any treatment and transfer of personal data. More concretely, if any personal data or identifiable information is being recorded with or without consent, the Organic Law 3/2018 of December 5 on Protection of Personal Data and Guarantee of Digital Rights would apply, which recognizes the right to limit and even suppress any unwanted associated data. To avoid falling into regulation territory, before the analysis of any resource, the intention is to use data anonymization techniques for the text in order remove any reference to individuals so personal details are out of our context. More over, the use of the data for the tests and development of this thesis is all public information displayed on the AI4EU website that anyone can get access, with the purpose of being distributed.

The philosophy is simple, the resource classification program and knowledge graph is not storing personal data while in transit from their main platform, denoting that only the provided resource data is being taken care of while all the personal and profile data stays on AI4EU's hands and under their privacy policy.

Appendix D

Budget

In this section I will discuss the economic cost of the project. To do so, I will describe the staff cost, specifying the roles needed, and the generic and indirect costs. Furthermore, I will explain the strategy to control potential budget deviations.

D.1 Staff costs

First of all, it is necessary to identify the personnel profiles that will intervene in the development of the project. Divided into several roles: the project manager, developer, researcher and tester. The responsibility of the project manager is shared between the director and the student, while the rest of the jobs are done individually by the latter. In the following table D.1, it is shown the estimated retributions in euros by the hour that corresponds to each of the profiles.

Role	**Cost (€/h)**
Project Manager	23
Junior Researcher	15
Junior Developer	22
Tester	8

Table D.1: Cost per hour of the different roles [39].

Given the definition of the roles needed and its cost, the next step is to define the relation between the Gantt chart (Figure 2-1) and the different roles. The table D.2

has been used to calculate the CPA (Cost Per Activity).

Id.	**Task**	**Hours**	**Project Manager**	**Developer**	**Researcher**	**Tester**	**Cost (€)**
T1	**Project planning and documentation**	110					2.530
T1.1	- Contextualization and project scope	24	24				
T1.2	- Temporal planning	8	8				
T1.3	- Economic management and sustainability	8	8				
T1.4	- Documentation	60	60				
T1.5	- Oral defense preparation	10	10				
T1.6	- Meetings	20	20				
T2	**Preliminary work**	30					450
T2.1	- FreeLing	10			10		
T2.2	- Ontology	10			10		
T2.3	- Corese	10			10		
T3	**Dev. of the Partial Knowledge Graph**	110					2.224
T3.1	- Design	12			12		
T3.2	- Implementation	90		90			
T3.3	- Validation	8				8	
T4	**Development of the Reasoning Module**	110					2.224
T4.1	- Design	12			12		
T4.2	- Implementation	90		90			
T4.3	- Validation	8				8	
T5	**Development of the Learning System**	110					2.224
T5.1	- Design	12			12		
T5.2	- Implementation	90		90			
T5.3	- Validation	8				8	
T6	**Integration and final evaluation**	50					890
T6.1	- Integration	30		30			
T6.2	- Final evaluation	20			10	10	
	Total						**10.542**

Table D.2: CPA Table

D.2 Generic costs

Amortizations

During the development of the project the only hardware needed will be the computer that I already own, which is a desktop that cost 1.300 € a couple of years ago. Taking into account that the lifespan of the device is of approximately 96 months (8 years), the amortization of the hardware is 8/96 x 1.300 = 108 €. In the case of the software used to program, as it is either free or open source, the amortization will be of 0 €.

Indirect costs

- Internet: The internet cost is of 38 €/month. Hence, the total cost would be (38 €/month) x (4 months) × (4,5/24 hours a day).

- Commute: In times of a global pandemic, the costs of transportation will be of 0 € replacing it with free virtual meetings, meaning the cost would be absorbed with the Internet payment.
- Electricity: The kWh price in my case is of about 0.12 €/kWh. Given that the average power of my desktop is of about 70 W, the total cost would be as follows (70 W) x (540 h) = 37.800 Wh or 37,8 kWh then (37,8 kWh) x (0,12 €/kWh).

Total of the generic costs

Below in the table D.3 I summarize the generic cost (CG) of the project, including the amortization of the hardware resources and the indirect costs.

Description	**Cost (€)**
Amortization	108
Internet	28,5
Commute	0
Electricity	4,5
Total	**141**

Table D.3: Total of the generic costs table

D.3 Budget deviations

Contingency

There is included a contingency margin of 10 percent for both the CPA and CG, which is in the realm of software development projects.

Incidental costs

Incident	**Estimated cost (€)**	**Risk (%)**	**Cost (€)**
Inability to adapt existing tools (30h)	450	70	315
Inexperience in the field (20h)	300	25	75
Ongoing platform (10h)	150	20	30
External factors	0	50	0
Total	900	-	**420**

Table D.4: Incidental costs table

Final budget

The last thing to do is estimating the final budget of our project by grouping all the previous sections.

Activity	**Cost (€)**
CPA	10.542
CG	141
Contingency margin	1.068
Incidental cost	420
Total	**12.171**

Table D.5: Final budget table

D.4 Management control

Even though I have defined the part of the budget reserved to incidences in the previous section, I have not described how we are going to be taken care of the potential budget deviations. Below there is a procedure to detect any possible alteration in the projects cost.

While doing each task described in the planning section, I will calculate its individual deviation from the estimated cost, with the help of the following formulas.

- Estimated cost: Estimated cost of the task.
- Real cost: Real cost of the task. Hence, I shall recalculate the CPA, CG, contingency and incidents for each one to check if any problem that may increment its cost. Doing so helps to identify which part of the task has been miss-estimated in order to re-plan.
- Deviation: Difference between the estimated cost and the real cost of the task.

Therefore, the deviation factor indicates how much the real cost varies from its estimation. Note that if the difference between the estimation and the real cost is positive means this is due to an overestimation of the task cost. There is the possibility of investing this extra money for other incidences. On the other hand, if the difference is negative, part of the contingency fund will be assigned to that task in order to cover the deviation.

Appendix E

Sustainability

E.1 Self assessment

The increment in pollution and waste caused by consumerism or the impact in peoples mental health of the technology products we use are a few samples that paired with the economical viability have an impact in the sustainability of a project. Assessing the sustainability of the Bachelor Thesis ensures that we consider the footprint it may have in the economic, social and environmental sides. Although the majority of companies in this day and age are made exclusively for an economical profit, we have to make a change and also focus on the social and environmental impact, which in most cases can even be complimentary. Most of the time overseen, the simple fact of answering the poll and questions below show a dedication to take the topic into account and evaluate if the choices we are making are worth for it. The idea of amortizations or costs aren't really new to me, hence I have had already incorporated them into the decisions taking, while environmental consciousness has also been an area of interest for me always. Other concepts like the equity it has in society are really new to me, the selfishness in humanity is sort of by default so considering others is not usually taken into account and needs to be deliberated. However, there is not any part of the sustainability that can be omitted because every side can be critical to the viability of a project.

E.2 Environmental

Regarding the PPP, have you thought about the environmental impact of your project? Have you considered minimizing this impact, by for example reusing resources?

Owing to the prototyping nature of this project, it is hard to estimate its environmental impact. As far as I'm concerned, the only point that would have some impact in the environment is the energy consumption, due to both the implementation part of the system development and the possible later integration in a live environment.

Regarding the live expectancy, how is it solved the problem you are trying to solve? Does your solution provide any improvement in the environmental impact?

The introduction of the human assisted AI component in principle would not provide a significant improvement in the environmental department, however, the system would have the ability to speed up user interaction, thus enabling more efficient workflows.

E.3 Economic

Regarding the PPP, have you estimated the impact that your project will have, including both human and material costs?

In the section D a detailed description the impact of this project is found, considering both human and material costs, and including the potential deviations that may appear during the development of the project.

Regarding the life expectancy, how is it solved the problem you are trying to solve? Does your solution provide any improvement economically?

As an improvement to the quality of the AI4EU's resource publication and categorizing, it would certainly have an impact in the European funded effort to improve its data classification models.

E.3.1 Social

Regarding the PPP, how do you think this project will enrich you personally?

This is the first time I immerse myself in a project of these characteristics, meaning it is also a big opportunity to put into practice all the knowledge learnt through these years and what will come.

Regarding the life expectancy, how is it solved the problem you are trying to solve? How do you think your solution will improve people's quality of life? Is there a real need of developing your solution?

As a matter of fact, there is the expectation that Knowledge Graphs will follow users knowledge as it evolves and deliver accurate responses to queries, solving the need of organizing the growing flood of new information.

Appendix F

Output queries (extract)

```
...
IoT
IoT_solution
solution_IoT
IoT_This
This_IoT
IoT_solution
solution_IoT
IoT_aiming
aiming_IoT
combination
combination_IoT
IoT_combination
machine
learning
learning_machine
machine_learning
learning_machine
machine_learning
```

```
algorithmic_program

algorithmic_program_learning

learning_algorithmic_program

algorithmic_program_algorithm

algorithm_algorithmic_program

algorithmic_program_learning

learning_algorithmic_program

algorithmic_rule

algorithmic_rule_learning

learning_algorithmic_rule

algorithmic_rule_algorithm

algorithm_algorithmic_rule

algorithmic_rule_learning

learning_algorithmic_rule

algorithm

algorithm_learning

learning_algorithm

algorithm_learning

learning_algorithm

manufacture

manufacture_industry

industry_manufacture

manufacture_algorithm

algorithm_manufacture

industry

industry_algorithm

algorithm_industry

logistic

logistic_industry
```

```
industry_logistic

logistic_industry

industry_logistic

...
```

Appendix G

Output suggestions

Here are collected the four types of output suggestions for the IoT model resource.

First semantic:

```
{"logistic": ["logistics industry", "logistics enterprise"],
"iot": ["smart transport", "intelligent automotive solution",
"smart transportation solutions", "smart transportation systems",
"smart transportation", "smart transportation system",
"smart transportation solution", "intelligent automotive solutions",
"machine to machines", "smart homes", "smart environment",
"machine-to-machine (m2m)", "smart home", "ambient intelligence",
"smart environments"], "machine-learning": ["rbf network", "
k-nn classifier", "vergence", "gabor feature", "radio frequency (rf)",
"bp networks", "sift feature", "transponders", "global feature",
"small sample size problems", "epc", "rbf networks", "eigenfaces",
"least square support vector machines", "training algorithms"],
"machine learning": ["rbf network", "k-nn classifier", "vergence",
"gabor feature", "radio frequency (rf)", "bp networks", "sift feature",
"transponders", "global feature", "small sample size problems",
"epc", "rbf networks", "eigenfaces", "least square support vector machines",
```

"training algorithms"], "sensor": ["decode-and-forward",
"decode-and-forward (df)", "human tracking", "multicast routing protocol",
"secure group communications", "excimer", "radio frequency (rf)",
"wireless sensor node", "aodv protocols", "panchromatic images",
"transponders", "demultiplexers", "smart transport", "epc",
"subaperture"], "sensor data": ["sensor data", "sensor systems",
"sensor readings", "sensor device"]}

Second semantic:

{"logistic": ["logistics industry", "logistics industry", "logistics enterprise"],
"sensor": ["sensor readings", "sensor readings", "sensor readings",
"sensor data", "sensor data", "sensor data", "sensor device",
"sensor device", "sensor device", "sensor systems",
"sensor systems", "sensor systems"]}

First similarity:

{"logistic": ["optimization problems", "machine-learning", "machine learning"],
"iot": ["machine-learning", "machine learning"],
"machine-learning": ["machine learning", "neural network",
"classifiers", "classifier", "deep learning", "image classification",
"classification process", "visual features", "verification",
"optimization problems", "optimization", "network architecture", "document images",
"cognitive process", "computer vision"],
"machine learning": ["machine-learning", "neural network", "classifiers",
"classifier", "deep learning", "image classification", "classification process",
"visual features", "verification", "optimization problems",
"optimization", "network architecture", "document images", "cognitive process",
"computer vision"], "sensor": ["network architecture", "computer vision",
"neural network", "machine-learning", "machine learning"],
"sensor data": ["network architecture", "machine-learning",

```
"machine learning", "neural network"]}
```

Second similarity:

```
{"machine-learning": ["machine learning", "neural network", "neural networks"],
"machine learning": ["machine-learning", "neural network", "neural networks"]}
```

Bibliography

[1] "Introducing the Knowledge Graph: things, not strings," May 2012. [Online]. Available: https://blog.google/products/search/introducing-knowledge-graph-things-not/

[2] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction," 03 2015.

[3] "AI4EU | AI4EU." [Online]. Available: https://www.ai4eu.eu/

[4] "Welcome | FreeLing Home Page." [Online]. Available: http://nlp.lsi.upc.edu/freeling/index.php/

[5] AI4EU, "Manual on platform knowledge modelling (deliverable d3.4)," 2020. [Online]. Available: https://www.ai4eu.eu/ai4eu-project-deliverables

[6] "DCMI: DCMI Metadata Terms." [Online]. Available: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

[7] "FOAF (ontology)," Feb. 2021, page Version ID: 1008330907. [Online]. Available: https://en.wikipedia.org/w/index.php?title=FOAF_(ontology)&oldid=1008330907

[8] "CSO - Portal." [Online]. Available: https://cso.kmi.open.ac.uk/home

[9] A. Anjos, L. El-Shafey, and S. Marcel, "Beat: An open-source web-based open-science platform," 04 2017.

[10] T. Llewellynn, S. Koller, G. Goumas, P. Leitner, G. Dasika, L. Wang, K. Tutschku, M. Fernández-Carrobles, O. Deniz, S. Fricker, A. Storkey, N. Pazos, G. Velikic, K. Leufgen, and R. Dahyot, "Bonseyes: Platform for open development of systems of artificial intelligence: Invited paper," pp. 299–304, 05 2017.

[11] "European Language Grid." [Online]. Available: https://www.european-language-grid.eu/metadata/

[12] S. Chowdhury, "Knowledge Graph," Jul. 2019. [Online]. Available: https://towardsdatascience.com/knowledge-graph-bb78055a7884

[13] P. A. Bonatti, S. Decker, A. Polleres, and V. Presutti, "Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371)," 2019. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10328

[14] L. Shao, Y. Duan, X. Sun, H. Gao, D. Zhu, and W. Miao, "Answering who/when, what, how, why through constructing data graph, information graph, knowledge graph and wisdom graph." [Online]. Available: https://ksiresearch.org/seke/seke17paper/seke17paper_79.pdf

[15] C. Jiang, S. Lu, and M. Jagersand, "Constructing dynamic knowledge graph for visual semantic understanding and applications in autonomous robotics." [Online]. Available: https://arxiv.org/vc/arxiv/papers/1909/1909.07459v1.pdf

[16] "Corese Web Server." [Online]. Available: http://corese.inria.fr/

[17] "pugixml.org - Home." [Online]. Available: https://pugixml.org/

[18] "Raptor RDF Parser Toolkit - Raptor RDF parser utility." [Online]. Available: https://librdf.org/raptor/rapper.html

[19] "Apache Marmotta - Home." [Online]. Available: https://marmotta.apache.org/

[20] "Vaticle." [Online]. Available: https://vaticle.com/

[21] "Graph Database Platform | Graph Database Management System | Neo4j." [Online]. Available: https://neo4j.com/

[22] "Cypher Query Language - Developer Guides." [Online]. Available: https://neo4j.com/developer/cypher/

[23] Node.js, "Node.js." [Online]. Available: https://nodejs.org/en/

[24] "Express - Node.js web application framework." [Online]. Available: https://expressjs.com/

[25] J. F. js.foundation, "jQuery." [Online]. Available: https://jquery.com/

[26] "OAuth 2.0 — OAuth." [Online]. Available: https://oauth.net/2/

[27] "WordNet | A Lexical Database for English." [Online]. Available: https://wordnet.princeton.edu/

[28] "Domain specific NLP | FreeLing Home Page." [Online]. Available: http://nlp.lsi.upc.edu/freeling/node/582

[29] "XPath," Jun. 2021, page Version ID: 1026762711. [Online]. Available: https://en.wikipedia.org/w/index.php?title=XPath&oldid=1026762711

[30] “neo4j-contrib/neovis.js,” Jun. 2021, original-date: 2016-04-06T17:56:25Z. [Online]. Available: https://github.com/neo4j-contrib/neovis.js

[31] asoroa, “asoroa/ukb,” Jun. 2021, original-date: 2012-06-27T07:56:22Z. [Online]. Available: https://github.com/asoroa/ukb

[32] “Neural Dependency Parser - FreeLing User Manual.” [Online]. Available: https://freeling-user-manual.readthedocs.io/en/latest/modules/dep_lstm/

[33] “Introduction - Neosemantics.” [Online]. Available: https://neo4j.com/labs//neosemantics/4.0/introduction/

[34] “Introduction - APOC Documentation.” [Online]. Available: https://neo4j.com/labs/apoc/4.1/introduction/

[35] “Graph Data Science Algorithms | Graph Data Analysis | Neo4j.” [Online]. Available: https://neo4j.com/product/graph-data-science-library/

[36] M. Needham and A. Hodler, *Graph Algorithms: Practical Examples in Apache Spark and Neo4j.* O’Reilly Media, 2019. [Online]. Available: https://books.google.es/books?id=UwIevgEACAAJ

[37] “Handlebars.” [Online]. Available: https://handlebarsjs.com/

[38] “Natural Language Toolkit — NLTK 3.6.2 documentation.” [Online]. Available: https://www.nltk.org/

[39] “PayScale - Salary Comparison, Salary Survey, Search Wages.” [Online]. Available: https://www.payscale.com