FRAMEWORK FOR KNOWLEDGE ACQUISITION, REPRESENTATION AND USAGE, LEARNING AND COLLABORATION BETWEEN AGENTS AND HUMANS IN INTERNET INFRASTRUCTURE DOMAIN

TECHNICAL REPORT

Juliao Braga[*]
Center for Mathematics, Computing, and Cognition
Federal University of ABC
Santo André, SP, BR
j@braga.net.br

Itana Stiubiener[†]
Center for Mathematics, Computing, and Cognition
Federal University of ABC
Santo André, SP, BR
itana.stiubiener@ufabc.edu.br

March 7, 2024

ABSTRACT

The focus of this research project is to consolidate some activities associated with components of the environment called Structure of Knowledge Acquisition, Use, Learning, and Collaboration (SKAU), proposed in a doctoral thesis [1, 2]. The main objective is to populate a Knowledge Base (KB) with knowledge from the repository of documents called Request For Comments (RFCs) from the RFC Editor. Additionally, the project intends to carry out the construction of domain ontologies with knowledge taken from related environments, such as, for example, the ecosystem of the Internet Infrastructure and nearby knowledge domains. Three tools are used to represent the knowledge to be captured: Protégé 5, OnTop, and the dictionary data structure, from programming languages. The knowledge base hosting the captured knowledge will be used by autonomous agents present and organized in a structure called Autonomous Architecture over Restricted Domains (A2RD), which collaborate with each other and with humans through a structure of inversely linked lists called IIBlockchain. The issues that will be resolved through primary studies and experiments refer to the capture and representation of knowledge from other unstructured bases and the representation of knowledge provided by Description Languages and improved by the alignment process of the various results obtained.

Keywords ontology · Internet infrastructure · knowledge base · ietf · irtf

1 Problem Statement

The Internet is a network of computer networks. Such computer networks are called Autonomous Systems (ASes) and have a unique identification, as can be seen in Figure 1. The ASes are interconnected through a complex telecommunications infrastructure, spread all over the earth and beyond. The operation of the ASes that use this telecommunications infrastructure is done by programs, that is, algorithms generally called protocols.

This project is oriented to study and research techniques and resources to adapt an environment to support intelligent agents with actions on domains of Autonomous Systems (ASes) or Routing Domains [3].

The agents, which are also algorithms, are organized in a layered model called Autonomous Architecture over Restricted Domains (A2RD), depend on a knowledge base (KB) and a collaboration environment (IIBlockchain), components

[*] ttp://lattes.cnpq.br/4008970012663480
[†] http://lattes.cnpq.br/4008970012663480

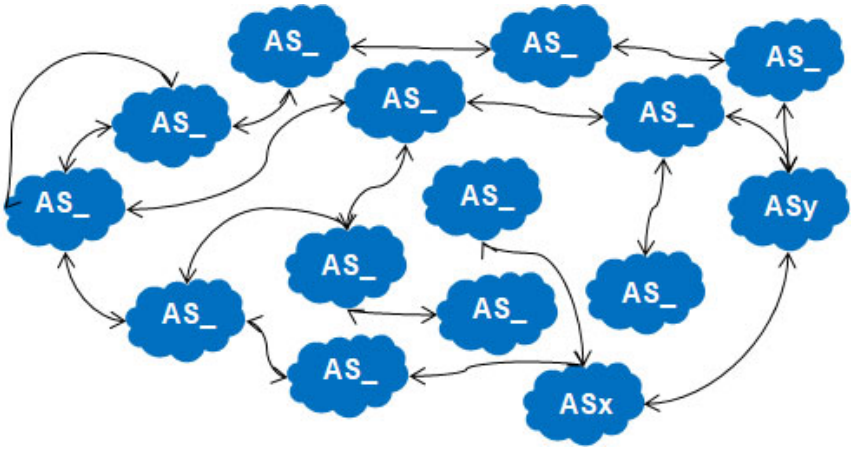

Figure 1: The autonomous networks that make up the Internet.

that belong to an ecosystem called Structure of Knowledge Acquisition, Use, Learning, and Collaboration (SKAU) [1], whose scenario can be seen in Figure 2.

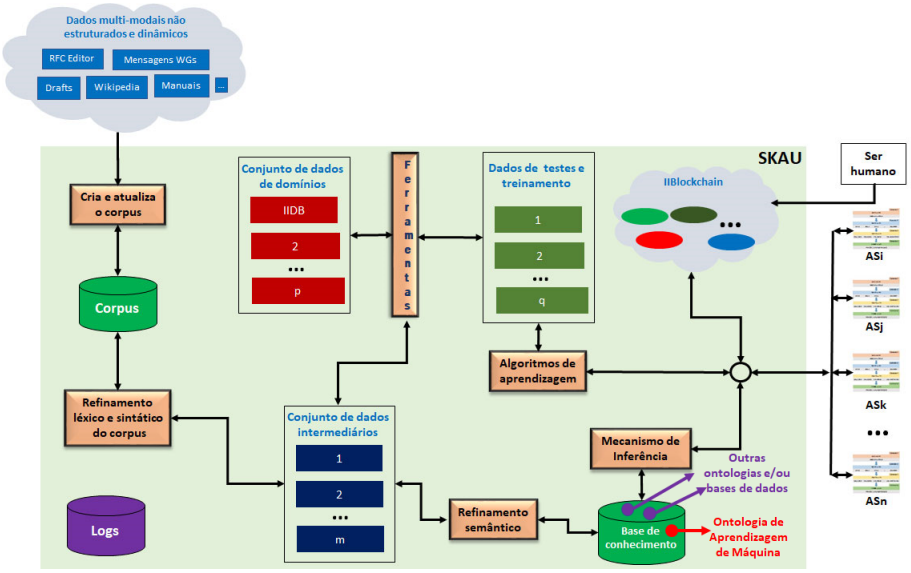

Figure 2: The SKAU environment, support for agents of the A2RD model, where the Knowledge Base is the main concern of this research. Adapted from [1].

The main focus of this research is the experience in filling the Knowledge Base (KB) with knowledge obtained: (i) from unstructured databases; (ii) from the manual and/or automatic construction of domain ontologies. The research is based on the application of Artificial Intelligence techniques and resources on the Internet Infrastructure, particularly those present in the area of Machine Learning (ML), with greater intensity, in the use of the facilities available in Natural Language Processing (NLP), whose relationship is visible in Figure 3 [4] [5] [6] [7] [8] [9].

The filling of the KB involves research questions related to the representation of knowledge captured from external (structured and unstructured) sources and with the techniques for storing this captured knowledge.

In addition, the agents of the A2RD model must improve their autonomy through cooperation among themselves and between human beings. The collaboration mechanism used by the agents is built on a database of inverted linked lists, called IIBlockchain [10]. The tooling for using the IIBlockchain is ready and will be intensively used, when appropriate, throughout the duration of the present research. The IIBlockchain was developed in Python, which will also be the main development language of the project.

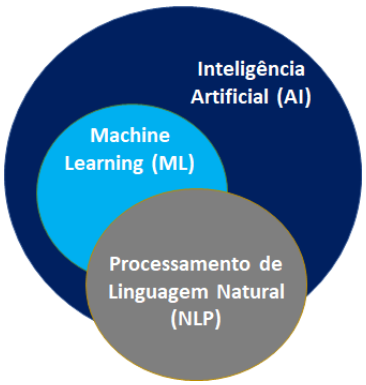

Figure 3: The relationship between AI, ML and NLP. Adapted from [9].

Besides this Introduction, named as Problem Statement (Section 1), this text exposes the Methodology that involves the project, in Section 3, with details of the techniques, resources, and facilities used. Section 4 indicates the results expected by the present research project, in the scheduled period, while Section 5 displays the challenges to be faced in view of the current stage of technology and the techniques involved. Section 6 clarifies how the results, the new proposals, and the lessons learned associated with the efforts undertaken in the project will be disseminated and evaluated.

2 The research environment

The main results expected for this research project will come from two activities of knowledge capture, each with multiple processes and linked to domains of the Internet Infrastructure. Such knowledge capture will occur from: unstructured bases and, the construction of domain ontologies. By ontology we mean a formal, explicit and shared representation of conceptual knowledge [11].

All data, algorithms, programs, and data are public domain and will be available in the project repository [12]. Following, the detailing of how the project will consider the activities in each of the alternatives of capture and representation of knowledge as well as the techniques and resources to be used [13].

2.1 Capture and representation of knowledge from unstructured databases

There are many unstructured bases of knowledge that are of interest to the project, and among them are those produced by the activities of the Internet Engineering Task Force (IETF), the Internet Research Task Force (IRTF), and associated organizations that orbit around it forming their own governance ecosystem, partially designed in Figure 4 [3].

Figure 2 displays, outside the SKAU environment, a set of unstructured and dynamic multimodal data, produced by the IETF, IRTF, and numerous other governmental, non-governmental, and private institutions, which are part of the ecosystem characterized above: RFC Editor, Messages generated by the Working Groups (WGs) of the IETF and IRTF, their preliminary texts (drafts), Wikipedia texts, product manuals and facilities, etc. In this project, during the first year, only the RFC Editor repository[3] will be used to form the knowledge base, with the capture of knowledge from the available documents [14]. Figure 5 illustrates the components involved in this process.

The documents, stored in the RFC Editor's repository, not named *Request for Comments* (RFCs) and their content represent the standards of protocols and the technical behaviors proposed to the administrators of Autonomous Systems (ASes), which allows the Internet to function better and better [3]. The processes that handle the RFCs can be simplified as represented in Figure 6.

Although the component **Inference Mechanism** remains in the figure, it will only be addressed during research associated with ontologies. It is a mechanism that, from the **Explicit Knowledge** defined by the rules of the ontology, can obtain the so-called **Implicit Knowledge**.

The use of a "nearly" structured repository like the RFCs (the natural language of the documents is structured and some groups of RFCs follow a pattern) makes the use of NLP easier to program and more efficient, as one can use a technique

[3]https://www.rfc-editor.org/about/

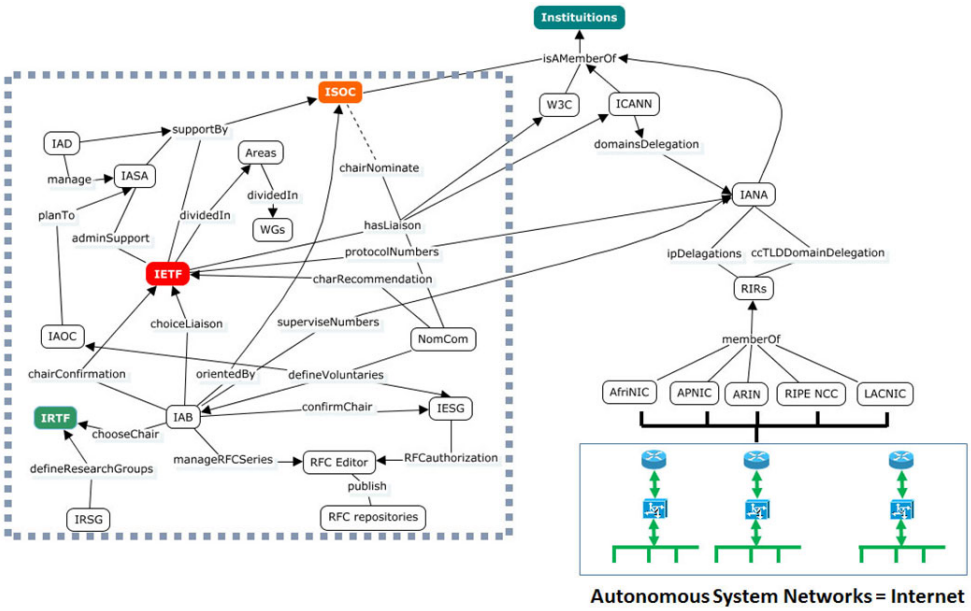

Figure 4: Internet Infrastructure Ecosystem with institutions around the IETF producing documents used by SKAU to partially populate the KB. Source: [1].

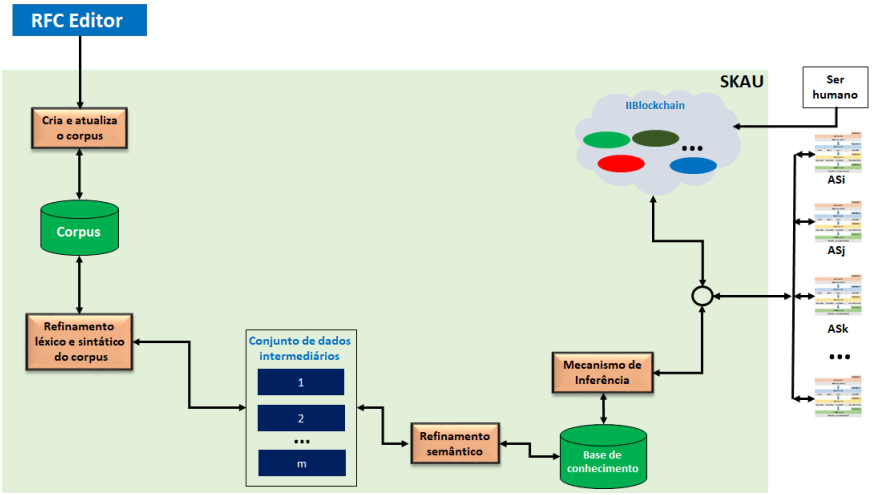

Figure 5: Interaction with the RFC Editor repository for knowledge capture and storage in the KB is the main focus of the present research. Adapted from [1].

of creating a **corpus** in stages called a *pipeline*, permanently (stored in digital media) or not, illustrated by Figure 7 showing the activities from the capture of the RFCs, to the update of the KB.

A simple example that would justify the creation of a corpus is the capture of acronyms and their meaning. In the RFCs, an acronym and its meaning have the following standard format:

$$Meaning\ of\ Acronym \quad (ACRONYM)$$

It is intuitive that the acronym and its meaning belong, completely, to a sentence. So, the creation of a corpus in which an RFC is represented by its distinguishable sentences, will make the processing of the search for acronym and its meaning quite efficient. A heuristic algorithm (using stacks) in which one looks for a "(" and a ")" within the sentence ensures that the acronym is what is between the parentheses. However, some acronyms are well formed, for example:

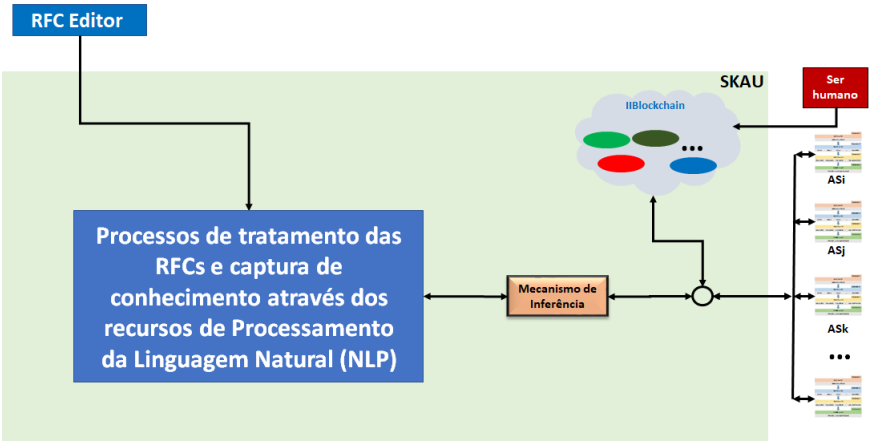

Figure 6: Core of RFCs processing. Adapted from [1].

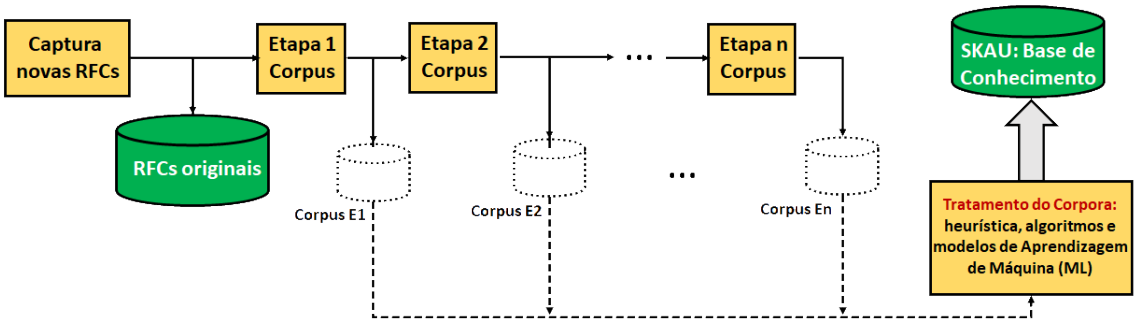

Figure 7: RFCs Corpora Treatment Pipeline

$Internet\ Engineering\ Task\ Force\ (IETF)$
$Internet\ Research\ Task\ Force\ (IRTF)$
$American\ Telephone\ and\ Telegraph\ (AT\&T)$

others are not so immediate:

$American\ National\ Standard\ Code\ for\ Information\ Interchange\ (ASCII)$
$IPv6\ over\ Low-Power\ Wireless\ Personal\ Area\ Networks\ (6LoWPAN)$

Figure 8 is the final representation of how RFCs will be treated when using the resources added to NLP.

2.2 Capture and representation of knowledge in domain ontologies

2.2.1 Preliminary

Ontology was defined in 1993 by Gruber and in 1997 was more appropriately improved by Borst [15] *apud* [16, 17]. In 1998, Studer and others adjusted the two definitions in the following proposal: "An ontology is a formal and explicit specification of a shared conceptualization" [18] [15]. Several authors refined the definitions over time to indicate, more clearly, that: **an ontology is a formally defined vocabulary for a particular domain, used to capture knowledge about this (restricted) domain of interest**. Therefore, an ontology describes the concepts of the domain and also the relationships that exist between these concepts [19]. The inclusion of a domain in the definition facilitated the understanding of the meaning of the ontology. In this way and finally, ontologies describe concepts and relationships in a specific domain, for Knowledge Representation (KR) and Knowledge Exchange (KE) [20].

On the other hand, **knowledge base** (KB) is a set of organized information about a certain subject. It can include facts, rules, procedures, and other types of knowledge. A knowledge base can be used to support a variety of applications, such as recommendation systems, question and answer systems, and artificial intelligence systems.

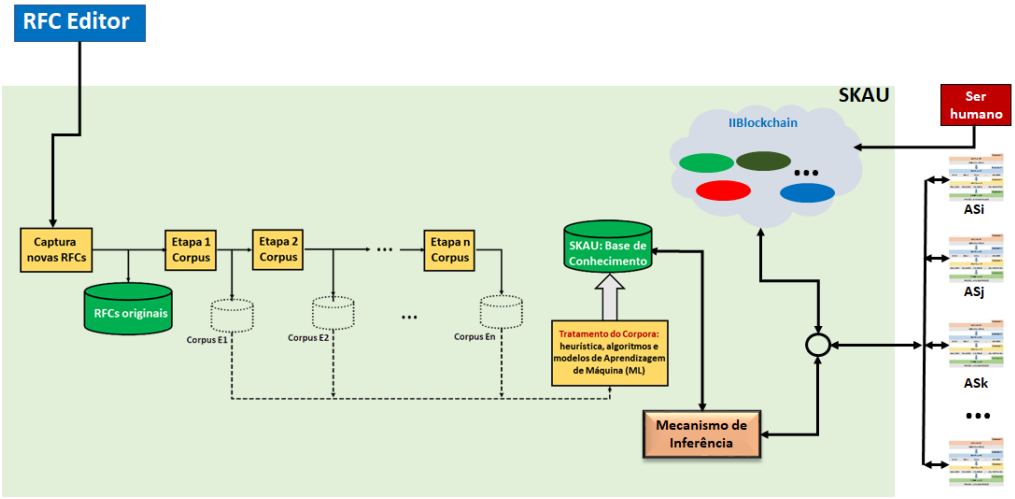

Figure 8: Components of the SKAU directly involved in the current project (replacing Figure 5), interaction with the RFC Editor repository for knowledge capture and storage in the KB, displaying the treatment of RFCs with the pipeline (Figure 7).

The main difference between ontology and knowledge base is that ontology is a formal model of a domain of knowledge, while the knowledge base is a set of organized information about a certain subject. The ontology is more abstract and generalist, while the knowledge base is more concrete and specific.

Another important difference is that the ontology describes the concepts and the relationships between these concepts, while the knowledge base can include facts, rules, procedures, and other types of knowledge. The ontology is, therefore, more relevant for the representation of structured knowledge, while the knowledge base is more relevant for the representation of unstructured knowledge.

The ontology and the knowledge base can be used together to represent the knowledge of a certain domain. The ontology provides the structure for the knowledge base, while the knowledge base provides the concrete data that are used to fill the structure.

Here are some examples of how ontology and knowledge base can be used together:

- A product recommendation system can use an ontology to represent the different types of products available, such as clothes, electronics, and food. The system's knowledge base can include information about the products, such as prices, reviews, and availability.
- A question and answer system can use an ontology to represent the different topics it can answer. The system's knowledge base can include information about these topics, such as definitions, examples, and facts.
- An artificial intelligence system can use an ontology to represent knowledge about the real world. The system's knowledge base can include information about the real world, such as the location of objects, the relationships between objects, and the events that occur in the world.

2.2.2 Knowledge Representation Languages

One of the most important concerns in building an ontology is how knowledge will be represented. In other words, this concern has to do with the formalism (or language) used to represent knowledge. Preferably this formalism should express the knowledge of a domain as completely as possible, without the possibility of inconsistencies occurring, especially when making inferences about the KB, of which the ontology will be part.

In some cases, whenever it is necessary to express knowledge using graphic tools or other less extensive representations, but with the aim of clarifying formal knowledge, the project will use the concepts around Description Logics (DLs), which represent a family of knowledge representation languages, which can be used to represent (or describe) the knowledge inserted in an application domain, in a structured and well-characterized form [21] [22].

Over the years, many languages have been proposed and researched, mainly under semantic precision of their reasoning procedures [23]. Initially, KL-ONE [24] emerged, which led to the emergence of K-REP [25], BACK [26], and LOOM [27], among others. All led to $\mathcal{SROIQ}$ [28], the DL that gave rise to the Web Ontology Language (OWL) 1 and,

finally, to OWL 2 [29]. Figure 9 shows the DLs derived from KL-ONE. The graph is the result of an ontology, still in development, of the DLs created to improve the representation of knowledge, in ontologies.

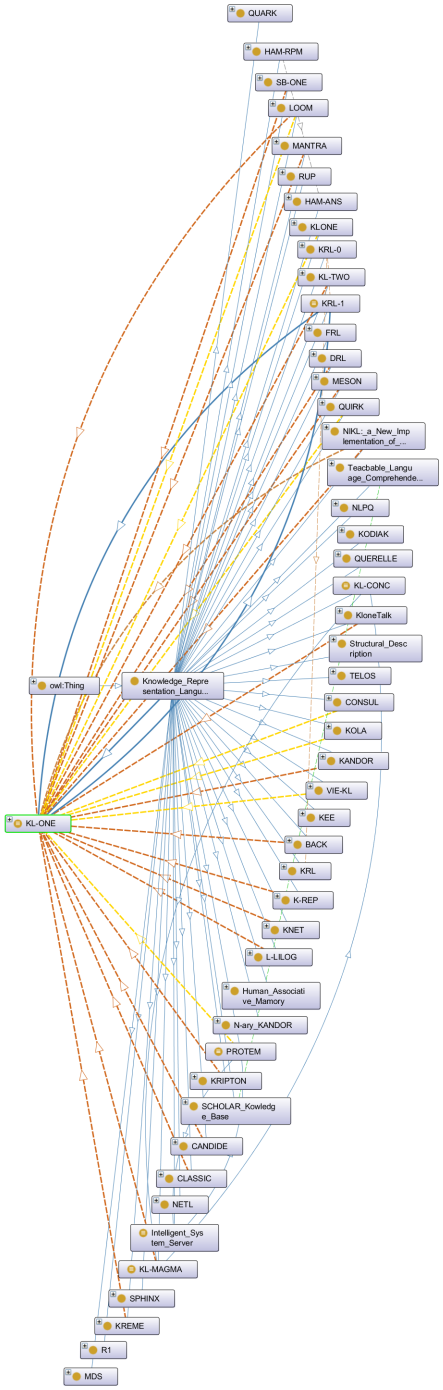

Figure 9: Partial illustration of Knowledge Representation Languages where most of them are derived from KL-ONE (represented separately) established in 1978. Graph obtained through the **OntoGraf** plugin, from Protégé.

2.2.3 The Language of the Project

OWL 2 [30, 31, 29], is part of the DLs family. It is the preferred language in the construction of ontologies, in the present project, as well as others, as long as they adhere to it, such as the *Resource Description Framework*[4] (RDF) and the *RDF Schema*[5] (RDFs).

3 Methodology

The project will be developed in two stages: (a) Manual and semi-automatic construction, when necessary, of a partial ontology in the domain of Internet Infrastructure and (b) Construction of an ontology, on the same domain, based on deep learning techniques.

The manual and/or semi-automatic ontology will be built through **Protégé**[6] in its version 5. Various techniques and algorithms can be used to make this partial ontology a model that displays an adequate experience for the second stage. Parallel to the works of the first stage, a systematic literature review will be developed in search of the most recent main recommendations on the techniques and algorithms (some described below) of deep learning available to be used in the second stage.

The ontology based on deep learning will be built using a combination of Natural Language Processing (NLP) techniques and machine learning. Initially, raw data will be collected from various reliable sources related to Internet Infrastructure and many of them identified by the manual and semi-automatic exercise of the first part. This data will then be pre-processed to remove noise and make them suitable for analysis.

Next, a deep learning model will be trained on this pre-processed data. The model will be configured to identify and learn the most important concepts and relationships in the domain of Internet Infrastructure. The result will be a rich and detailed ontology that captures knowledge in this domain in a structured and accessible way.

Finally, the two ontologies - the manual and the one based on deep learning - will be integrated to form a complete ontology. This combined ontology will then be validated by domain experts to ensure its accuracy and usefulness.

3.1 Some Deep Learning Techniques and Algorithms

The choice of deep learning technique will depend heavily on the systematic literature review and the data collected and to be processed. It is possible that several techniques will be experimented with, among which those that offer the best prospects for aligning with the project's objectives will be chosen.

There are several deep learning techniques that can be used for the construction of the ontology, depending on the specific data and the needs of the project. Here are some possibilities:

1. **Convolutional Neural Networks (CNNs)**: They are especially effective in processing visual data and can be used if the data includes images or diagrams related to Internet Infrastructure. It is known that these are the cases of the Request for Comments (RFCs), available in the RFC Editor's repository[7]. Examples of these algorithms are: LeNet[8], AlexNet[9], VGG[10] (Visual Geometry Group), GoogLeNet[11], and ResNet[12].
2. **Recurrent Neural Networks (RNNs)**: They are useful for dealing with sequential or temporal data. If the data to be collected have a temporal component (for example, time series of network traffic), RNNs can be a

[4] `https://www.w3.org/TR/rdf-concepts/`

[5] `https://www.w3.org/TR/rdf-schema/`

[6] https://protege.stanford.edu/

[7] `https://rfc-editor.org.br`

[8] Generally referring to LeNet-5, it is a simple CNN that stands out in the processing of large-scale images and was one of the first convolutional neural networks to boost the development of deep learning (1998).

[9] It is a CNN architecture, which was designed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton. It stood out in the *ImageNet Large Scale Visual Recognition Challenge* on September 30, 2012.

[10] It is a standard CNN architecture, with several layers, being known for its versions VGG-16 and VGG-19, which consist of 16 and 19 convolutional layers, respectively.

[11] Also known as Inception V1, it is a CNN architecture proposed by Google in 2014, which uses *Inception* modules to choose between various sizes of convolutional filters in each block, allowing the creation of deeper architectures

[12] It is a deep learning model that uses identity connections (or "residual connections") to allow the effective training of deep networks with tens or hundreds of layers, improving accuracy by increasing depth.

good choice. Among others: LSTM[13] (Long Short-Term Memory) and GRU[14] (Gated Recurrent Unit) are examples of RNNs.

3. **Transformers**: These are deep learning models that use attention mechanisms to improve effectiveness in processing sequential data. They have been widely used in Natural Language Processing (NLP) tasks. BERT[15] (Bidirectional Encoder Representations from Transformers), GPT[16] (Generative Pretrained Transformer), and T5[17] (Text-to-Text Transfer Transformer) are examples of Transformers.
4. **Autoencoders**: These are neural networks used to learn encoded (or compact) representations of input data. They can be useful for learning the latent structure of the data to be collected. VAE[18] (Variational Autoencoder) and *Denoising Autoencoder*[19] are examples of Autoencoders.
5. **Generative Adversarial Networks (GANs)**: These are pairs of neural networks that work together to continuously improve the quality of eventual predictions. Although they are best known for their ability to generate realistic images, they can also be used in other types of data. Examples of GANs: DCGAN[20] (Deep Convolutional GAN), CycleGAN[21] and StyleGAN[22].

4 Expected Results of the Project

A Populate the knowledge base (KB) of SKAU with knowledge captured from the RFCs;
B Populate the KB of SKAU with knowledge obtained through the construction of ontologies of domains that interest the Agents of the A2RD model;
C Increase the KB with domain ontologies built by third parties and qualitatively aligned with those already existing;
D Propose development standards for RFCs that facilitate the use of ML techniques and algorithms. For example, propose standards for creating acronyms avoiding ambiguities that may create problems for ML techniques and algorithms;
E Document and disseminate the treatment of repositories associated with the knowledge of a restricted domain;
F Develop a process of knowledge collection to be done by the agents of the A2RD model. For example, the RFC repository could be worked on by the agents without human intervention. Initially, the agents would build ontologies with knowledge mined from the RFCs. A good example is acronyms. From the general form of an acronym in the RFCs, which would be discovered by the agents themselves, an ontology of acronyms would be built and automatically populated in the KB. The expected result in this direction of automation will be intermediate, that is, it is expected that the experience will identify the ways to make the agents autonomous in the construction of ontologies.
G Establish a quantitative efficiency analysis of the ontologies created using DLs, via Protégé for example, the results by using OnTop and the knowledge stored using Dictionaries. Additionally, qualitative analyses of the use of these resources from the human point of view, as a user, will be included. (**The data structure of the dictionary and the use of databases stored in JSON will be intensively used, to make the main results of the research available, for third-party use.**)

[13] It is a recurrent neural network (RNN) that can process data sequentially and maintain its hidden state over time, being designed to deal with the problem of the disappearance of the gradient present in traditional RNNs.

[14] It is a variant of the recurrent neural network (RNN) that uses update and reset gates to solve the problem of the disappearance of the gradient, making it more effective in processing long sequences.

[15] It is a deep learning model that improves the efficiency of natural language processing (NLP), being famous for its ability to consider context when analyzing the relationships between words in a sentence in a bidirectional way.

[16] It is a deep learning model that uses transformers to generate complex mathematical representations of text or other types of media, allowing a computer to perform some tasks in a way similar to the human brain.

[17] It is a deep learning model that transforms all language tasks into a text-to-text task, allowing the model to learn more effectively from unlabeled data.

[18] It is a neural network architecture that compresses the input into a latent space representation and then reconstructs the output from this representation, being effective for learning latent representations.

[19] It is a type of artificial neural network that learns to encode data efficiently in an unsupervised manner, with the main goal of learning a representation (encoding) for a set of data, introducing a reconstruction constraint, and has the ability to reconstruct the input from a corrupted version, effectively learning to remove noise from the data.

[20] It is a generative adversarial network architecture that uses strided convolutions and fractional convolutions to replace pooling layers, as well as using batch normalization (batchnorm) in both the generator and the discriminator.

[21] It is a technique that allows translation between two different domains without the need for paired data, using cycle consistency and adversarial training to learn mappings between two domains using unpaired data.

[22] It is a generative adversarial network (GAN) that uses an alternative architecture to generate high-quality synthetic images, offering control over the style of the generated image at different levels of detail.

H With attention focused on NLP techniques and the linguistic specificities characterized by the corpus repository managed by the RFC Editor, it can be established that facilities could be implemented or indicated to optimize the work of producing drafts and/or RFCs, avoiding the not very friendly tools available today [3]. Also, a search mechanism in RFCs could be implemented, very efficiently, by recommending the next words in a given sequence or even a sentence or text, in addition to facilitating the study of RFCs oriented to a specific subject [32] [33] [34].

I Results of exercises on **language models** (including variations of **N-grams**[23] confronted with evaluation measures of **perplexity**[24]), in the RFC Editor repository could bring other results not yet thought at this moment.

J It is expected to have a comprehensive and significant attention in this research in scenarios that require exploiting **knowledge graphs** [35]. We hope the project will contribute future research in this directions.

4.1 Dissemination and Evaluation

In addition to works that will be submitted to events associated with the theme, two seminars will be held promoted by the institutions involved in the project. The first seminar will have as its main theme **the construction of manual or semi-automated ontologies**. The second seminar will have as its central theme **the use of deep learning in the contraction of ontologies in the domain of Internet Infrastructure**.

4.2 Schedule

Stage 1 corresponds to two activities (A1.1 and A1.2) that will be executed in parallel and are, respectively: (A1.1) manual and/or semi-automatic construction of an ontology for Internet Infrastructure and (A1.2) systematic literature review on the application of deep learning in ontology. Stage 2 only one activity (A2.1), corresponding to the application of deep learning in the construction of an ontology.

Still in Stage 1, two articles are scheduled to be delivered. The first (E1.1) refers to the experience of manual and/or semi-automatic construction of the proposed ontology and the second (E1.2) refers to the result of activity A1.2, that is, the result of the systematic literature review.

Activity A2.1 will have as deliverables, two articles to be made available to publications (E2.1 and E2.2).

In each semester of the project, a technical report is planned on the results of the semester and the lessons learned up to the time of its publication (R1, R2, R3, and R4).

At the end of each year, seminars (S1 and S2) will be promoted for dissemination and evaluation of the state of the art of the two stages of the project.

Complementary to the project, a record of all its activities will be documented in the project repository, hosted on OSF (*Open System Framework*), through documents, especially the SKAU-CODEX as recorded in Figure 14 and other descriptions made in Section 6 [36].

In Table 1, the visual schedule of activities.

5 Scientific and Technological Challenges and the Techniques and Methods that Can Be Used in the Project

Two technological challenges must be considered: (a) **knowledge representation** and (b) **knowledge discovery**.

Both are complex challenges, but the first one is less than the second. This conclusion is drawn from the literature and the researcher's experience in dealing with these two issues. Programming languages, by providing a vast diversity of data structures, such as the **dictionary**, for example, help to construct the proposals for **knowledge representation** that science recommends.

However, the scientific recommendations for computationally treating knowledge representation, as well as knowledge discovery, are characterized by paradigms and derived proposals that make the choice difficult.

The solutions to face these two challenges are intense dedication to primary studies and experience (implementation of choices), about which we speak in the two subsections that follow.

[23] Popular name given to Markov Chain

[24] It is a measure that is used to evaluate prediction models. The lower the perplexity, the better the model

Table 1: Execution schedule of activities per semester.

Activities	1^o Sem.	2^o Sem.	3^o Sem.	4^o Sem.
A1.1 (Manual Ontology)				
A1.2 (Systematic Review)				
A2.1 (Deep Learning)				
E1.1 (Article: Manual Ontology)				
E1.2 (Article: Systematic Review)				
E2.1 (Article: Deep Learning)				
E2.2 (Article: Deep Learning)				
R1 (Technical Report)				
R2 (Technical Report)				
R3 (Technical Report)				
R4 (Technical Report)				
S1 (1st Dissemination and Evaluation Seminar)				
S2 (2nd Dissemination and Evaluation Seminar)				

5.1 Knowledge Representation

The challenge of representing knowledge has a complement that precedes it, known as **capture** of this knowledge. In the research being proposed, the domain of knowledge is narrow, when one thinks about the context of the RFC Editor repository. This fact facilitates the activity involved in this issue and one can start from the following primary studies to search for and experiment with an acceptable technical alternative: [37] [38] [39] [40] [41] [13] [42] [43] [44]

5.2 Knowledge Discovery

This second challenge is called **knowledge discovery from the RFCs repository** (in English, *knowledge discovery from RFCs* or KDR).

The primary studies related to this challenge are many and the paradigms are quite diversified. Despite the domain of knowledge contained in the RFCs being quite narrow, the effort for a choice or choices that disappear with the challenge depends on an extensive availability of academic articles. Since the end of 2019, a series of information has been annotated in the SKAU_UFABC Codex from a non-systematic literature research, which listed about 62 (sixty-two) primary studies from which alternatives could be evaluated, Table 2.

Table 2: Levantamento Preliminar de Estudos Primários em Sumarização e Extração de Dados

#	Estudos Levantados	2019	2021
1.	[45]	3675	4095
2.	[46]	26430	30470
3.	[47]	1865	2038
4.	[48]	420	456
5.	[49]	5632	6237
6.	[50]	379	404
7.	[51]	-	77
8.	[52]	2732	3099
9.	[53]	463	509
10.	[54]	1369	1479
11.	[55]	-	1813
12.	[56]	21	237
13.	[57]	112	112
14.	[58]	29	33
15.	[59]	417	490
continua na próxima página			

Table 2 – continuação da página anterior

#	Estudos Levantados	2019	2021
16.	[60]	1813	601
17.	[61]	35	36
18.	[62]	3977	6093
19.	[63]	1571	1818
20.	[64]	4	45
21.	[65]	-	166
22.	[66] RT	-	-
23.	[67]	395	444
24.	[68]	456	545
25.	[69]	234	277
26.	[70]	-	629
27.	[71]	3	4
28.	[72]	-	570
29.	[73]	40	51
30.	[74]	-	239
31.	[75]	-	501
32.	[76]	76	87
33.	[77]	14	16
34.	[78]	2	3
35.	[79]	-	12
36.	[80]	-	-
37.	[81]	-	-
38.	[82]	-	-
39.	[83]	-	-
40.	[84]	10	22
41.	[85]	85	119
42.	[86]	15	25
43.	[87]	40	193
44.	[88]	5	23
45.	[89]	1	-
46.	[90]	4	6
47.	[91]	-	393
48.	[92]	-	9
49.	[93]	-	20
50.	[94]	-	31
51.	[95]	-	17
52.	[96]	-	17
53.	[97]	-	134
54.	[98]	-	50
55.	[99]	-	17
56.	[100]	-	24
57.	[101]	1	-
58.	[102]	2167	3377
59.	[103]	-	1
60.	[104]	-	1
61.	[105]	-	-
62.	[106]	1	5

Details of each article are summarized in the aforementioned Codex. These annotations will be available in the project repository.

The two columns in Table 2 represent the citations received by each article at the end of 2019, when the survey was conducted, and at the time of 2021, when the project was presented. From the citations of the primary studies, it is possible to reach more recent studies. This search must be extensive and criteria will be adopted (and recorded), so that there is enough time for the necessary conclusions.

For the challenges described earlier, it should be kept in mind that most of the experiments will be focused on ML algorithms [7] [8] [107]. If, due to the short amount of time, it is not possible to implement all the desired experimentation, detailed documentation about it will be exposed in the SKAU_UFABC Codex to, eventually, be implemented in future opportunities.

5.3 Ontology Alignment

The research activities that involve ontologies are mainly associated with the problem of ontology alignment. It refers to the transformation of two or more ontologies into a single ontology, where the similarities are properly adjusted [108]. An intuitive view can be seen in Figure 10.

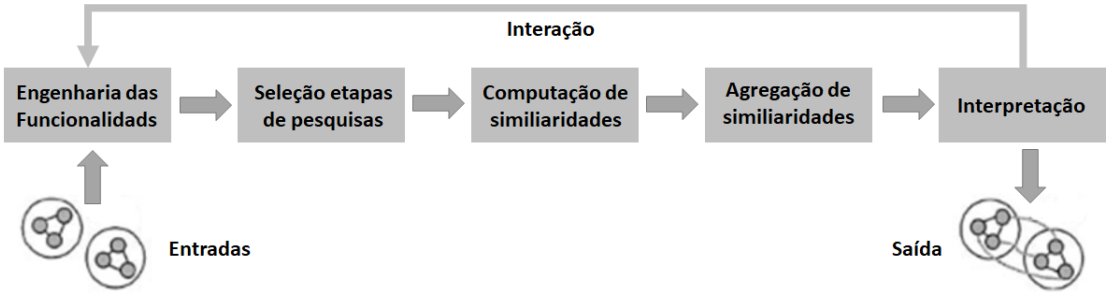

Figure 10: The input, exit, and the six main steps involved in the alignment process. Adapted from [109]

Experiments with ontology alignment will expand the perspective of populating the knowledge base, thus strengthening the expectation of knowledge available to the agents of the A2RD model. This is shown in Figure 11.

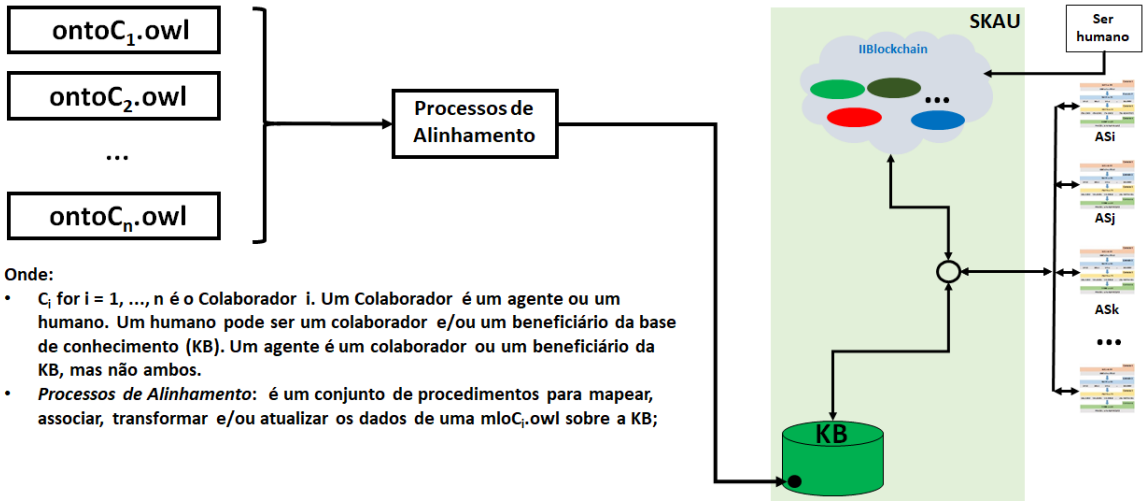

Figure 11: How domain ontologies are inserted into the knowledge base through alignment processes

5.4 OBDA

Ontop is a system that implements OBDA, with an architecture similar to that shown in Figure 12 to build virtual knowledge graphs [110, 111, 112],

Instead of structuring the integration layer as a collection of relational tables, ONTOP is structured as a Virtual Knowledge Graph (VKG) [113]. According to the same author, the VKG approach combines three ideas, brought below via a free translation:

VKG.1 Data virtualization (DV) [114] [115] [116], all cited by [113] obtained by avoiding exposing end users to real data sources, and instead presenting them with a conceptual representation of the domain of interest, typically called a global schema.

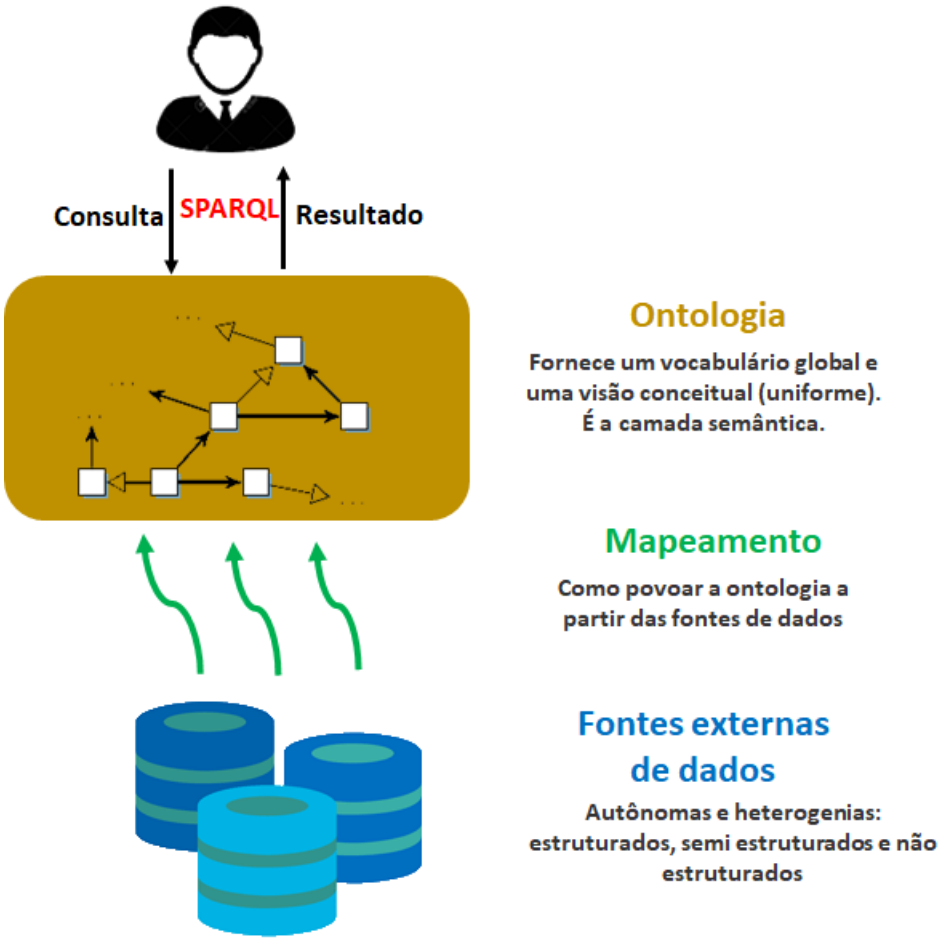

Figure 12: What is OBDA

VKG.2 The data is structured in the form of a graph (G), where domain objects and data values are represented as nodes. The properties of the objects are encoded as edges [117].

VKG.3 The graph that represents the data is enriched by domain knowledge (KD), capturing, for example, hierarchies of concepts and properties, domain and range of properties, and mandatory properties [118] [119]. Such knowledge allows to infer about the data and knowledge, and thus derive new implicit knowledge from the explicitly stated.

ONTOP makes extensive use of SPARQL [120] and has an integration with Protégé.

5.5 Apache Jena

The implementation of Apache Jena[25] will allow for further experimentation with SPARQL[26] and ARQ[27], participating in graph representation, of RDF[28]. Apache Jena uses SQL, allows integration with HTTP (Fuseki) and supports the use of OWL to build ontologies.

The development base of Jena is Java. Jena will be one of the last tools to be analyzed, only if none of the others satisfy the research.

[25] https://jena.apache.org
[26] https://jena.apache.org/tutorials/sparql_pt.html
[27] https://jena.apache.org/documentation/query/manipulating_sparql_using_arq_pt.html
[28] https://jena.apache.org/tutorials/rdf_api_pt.html

5.6 Ontology in Python

As Python is the programming language chosen for the project, the available libraries will be explored whenever necessary and possible. Python, as a dynamic language, has offered resources and facilities to handle ontologies. The **RDFLib** offers several methods to extract information from RDF graphs [121]. More recently, the **Owlready** library is proving to be suitable for handling ontologies with great ease via Python, including databases, similar to Ontop [122] [123].

Other facilities of the Python language in ontologies will be discovered and documented in the project repository.

5.7 Computational Linguistics

Computational Linguistics is a multidisciplinary field that uses techniques of Natural Language Processing (NLP) involving Artificial Intelligence, mainly its sub-area Machine Learning (ML), applied to the area of Linguistics [124].

The corpora maintained by the RFC Editor are part of a very appropriate and restricted domain, for experiments produced by **language models**, among other techniques and recommendations based on computational linguistics [125], general linguistics [126] and in particular, computational cognitive science [127].

There are many NLP techniques that will be explored and documented in detail in this research project. There are plenty of references available and related to NLP using Python, mainly addressing the linguistic structure of English [128] [129] [130].

5.8 Natural Language Processing

Processing the corpora managed by the RFC Editor (with more than 9,000 documents) may require a very large computational effort and a very high memory capacity. For example, in Markov models (n-grams) the size grows exponentially. Assuming a vocabulary V: a bigram model the size will be V^2; a trigram model, V^3; a quadrigram model, V^4; and so on [131].

Such challenges could be faced with changes in the memory demands of the algorithms increasing the processing cost, which would require computational power. Alternatives for the use of powerful computational resources available in Brazil, such as the Petaflop Computing System of SINAPAD[29].

At the appropriate time, the alternatives will be evaluated, pointing out the suitable and available solution.

5.9 Normalization and Alignment of Ontologies produced by various tools

This is a challenge that arises when using tools to produce ontologies. The results can be different, even in the same domain. A process of normalization and alignment adapts the results before they go on to populate the KB container. It is assumed that, eventually, it may be possible to use other tools, still unknown. Figure 13 illustrates this process.

6 Record of Lessons Learned During the Development of the Project

The lessons learned during the research and development period will be recorded in a set of documents that will be part of the so-called SKAU_CODEX, where an extract can be seen in Figure 14. In due course, this material will be compiled into dissemination texts such as tutorials to be used by students and researchers in the respective areas. These documents may be published on the Internet Infrastructure blog[30], in the form of *preprints*, on the Wiki of the SKAU-UFABC project repository [36], in the project repository (SKAU Project Repository). The project members will always be available to provide lectures, interviews, or other forms of dissemination explicitly related to the project.

Finally, information about the progress of the project and its main results will be systematically available on Twitter, LinkedIn, and Instagram. Occasionally, other digital media channels may be used.

References

[1] Juliao Braga. *Ambiente para Aquisição de Conhecimento por Agentes em Domínios Restritos na Infraestrutura da Internet*. PhD thesis, Instituto Superior Tecnico & Universidade Presbiteriana Mackenzie, 2019. DOI: 10.31237/osf.io/nzmtf.

[29] `https://lncc.br/sinapad/`

[30] `https://ii.blog.br`

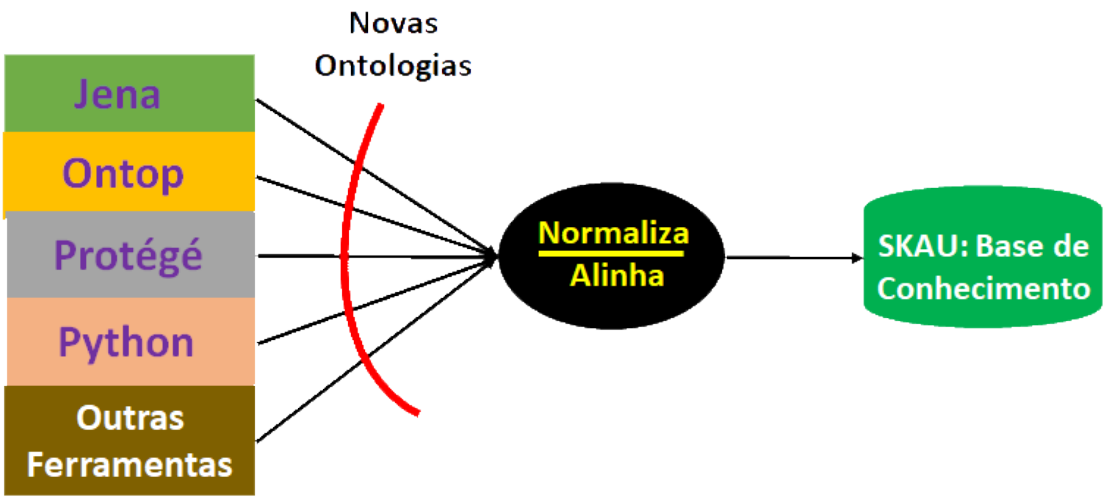

Figure 13: How the various results of ontologies produced by different tools are treated before being stored in the KB

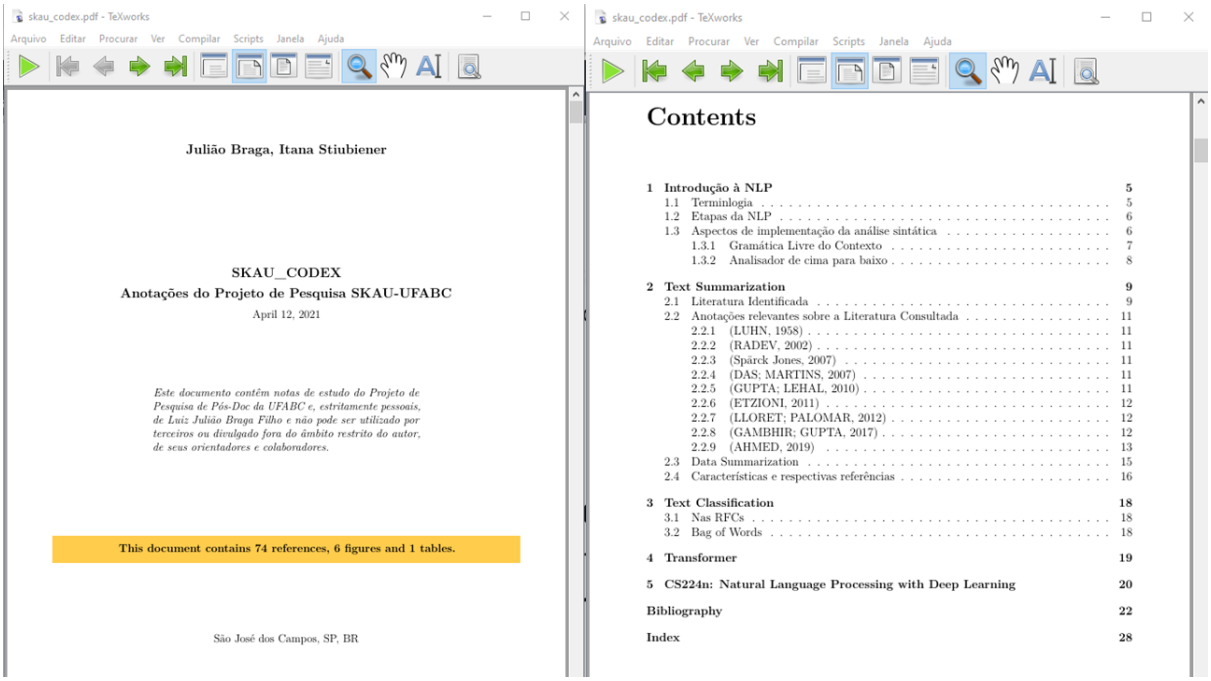

Figure 14: SKAU_CODEX: extract of the document. Source: Authors

[2] Juliao Braga. *Environment for Knowledge Acquisition by Agents in Internet InfrastructureRestricted Domains*. PhD thesis, Instituto Superior Tecnico & Universidade Presbiteriana Mackenzie, 2019. DOI: 10.31237/osf.io/83ztf. English version translated by author. Available in https://thesiscommons.org/83ztf/.

[3] Juliao Braga, Jeferson Campos Nobre, Lisandro Zambenedetti Granville, and Marcelo Santos. Como Protocolos Inovadores são Criados e Adotados em Escala Mundial: Uma visão sobre o Internet Engineering Task Force (IETF) e a Infraestrutura da Internet. In Taisy Silva Weber and Claudia Aparecida Martins, editors, *Jornadas de Atualiza cão em Informática 2020*, page 45. Sociedade Brasileira de Computa cão, Cuiabá, MT Brazil, 2020. Available in: https://doi.org/10.5753/sbc.5728.3.2.

[4] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[5] Stuart Russel and Peter Norvig. *Artificial Intelligence*. Prentice Hall, New York, 3 edition, 2010.

[6] Krishna Bhavsar, Naresh Kumar, and Pratap Dangeti. *Natural Language Processing with Python Cookbook: Over 60 recipes to implement text analytics solutions using deep learning principles*. Packt Publishing, 2017. ISBN: 97817872893211.

[7] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learning with Keras & TensorFlow*. O'Reilly Media, Inc., Canada, 2 edition, 2019. ISBN: 978-14920326491.

[8] Jens Albrecht, Sidharth Ramachandran, and Christian Winkler. *Blueprints for Text Analytics Using Python*. O'Reilly Media, Inc., New York, 1 edition, 2020.

[9] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*. O'Reilly Media, 2020.

[10] Juliao Braga, Joao Nuno Silva, Patricia Takako Endo, Jessica Ribas, and Nizam Omar. Blockchain to improve security, knowledge and collaboration inter-agent communication over restrict domains of the internet infrastructure, with human interaction. *Brazilian Journal of Development*, 5(7):9013–9029, july 2019. DOI:10.34117/bjdv5n7-103, ISSN 2525-8761.

[11] Jiaoyan Chen, Yuan He, Yuxia Geng, Ernesto Jiménez-Ruiz, Hang Dong, and Ian Horrocks. Contextual semantic embeddings for ontology subsumption prediction. *World Wide Web*, pages 1–23, 2023. DOI: `https://doi.org/10.1007/s11280-023-01169-9`.

[12] Juliao Braga and Itana Stiubiener. Skau-ufabc, Apr 2021.

[13] Sanida Omerovic, VELJKO MILUTINOVIC, and SASO TOMAZIC. Concepts, ontologies, and knowledge representation, 2001.

[14] Michael Gelfond and Yulia Kahl. *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press, 2014.

[15] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer, 2009.

[16] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[17] Willem Nico Borst. *Construction of engineering ontologies for knowledge sharing and reuse*. PhD thesis, University of Twente, 1997.

[18] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1):161–197, 1998.

[19] Mathew Horridge. A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3, 2011. `http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf`. Acessado em 31/03/2015.

[20] Thabet Slimani. Ontology development: A comparing study on tools, languages and formalisms. *Indian Journal of Science and Technology*, 8(24):1–12, 2015.

[21] Franz Baader, Ian Horrocks, Lutz Carsten, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[22] Franz Baader, Diego Calvanese, Deborah McGuinness, Peter Patel-Schneider, Daniele Nardi, et al. *The description logic handbook: Theory, implementation and applications*. Cambridge University Press, 2003.

[23] William A. Woods and James G. Schmolze. The kl-one family. *Computers Mathematics with Applications*, 23(2):133–177, 1992.

[24] Ronald J Brachman and James G Schmolze. An overview of the kl-one knowledge representation system. *Readings in artificial intelligence and databases*, pages 207–230, 1989.

[25] Eric Mays, Robert Dionne, and Robert Weida. K-rep system overview. *ACM SIGART Bulletin*, 2(3):93–97, 1991.

[26] Christof Peltason. The back system—an overview. *ACM SIGART Bulletin*, 2(3):114–119, 1991.

[27] Robert Mac Gregor. The evolving technology of classification-based knowledge representation systems. In *Principles of semantic networks*, pages 385–400. Elsevier, 1991.

[28] I Horrocks, O Kutz, and U Sattler. The even more irresistible sroiq sroiq. In *Proceedings 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 57–67, 2006.

[29] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. Owl 2: The next step for owl. *Journal of Web Semantics*, 6(4):309–322, 2008. DOI: https://doi.org/10.1016/j.websem.2008.05.001.

[30] W3C-OWL. OWL 2 web ontology language document overview (second edition). Technical document, W3C, dec 2012. https://www.w3.org/TR/2012/REC-owl2-overview/.

[31] Deborah L. McGuinness and Frank van Harmelen. Owl web ontology language overview, 2004. W3C. http://www.w3.org/TR/owl-features/. Acessado em 19/02/2015.

[32] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[33] Eric Fosler-Lussier. Markov models and hidden markov models: A brief tutorial. *International Computer Science Institute*, 1998.

[34] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[35] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54, 7 2021.

[36] Juliao Braga. Skau project repository, Sep 2022. Acessed: https://osf.io/tka9u. DOI: https://doi.org/10.17605/OSF.IO/TKA9U.

[37] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021.

[38] Claudio Gutierrez and Juan F Sequeda. Knowledge graphs. *Communications of the ACM*, 64(3):96–104, 2021.

[39] Meenakshi Malhotra and TR Gopalakrishnan Nair. Evolution of knowledge representation and retrieval techniques. *International Journal of Intelligent Systems and Applications*, 7(7):18, 2015.

[40] Simon Kendal and Malcolm Creen. *An Introduction to Knowledge Engineering*. Springer, London, 1 edition, 2007.

[41] Shelley Powers. *Practical RDF: solving problems with the resource description framework*. " O'Reilly Media, Inc.", 2003.

[42] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *AI magazine*, 14(1):17–17, 1993.

[43] Brian C Vickery. Knowledge representation: a brief review. *Journal of documentation*, 1986. DOI: https://doi.org/0.1108/eb0267901.

[44] John Mylopoulos and Hector Levesque. An overview of knowledge representation. *GWAI-83*, pages 143–157, 1983.

[45] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[46] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297. Oakland, CA, USA, 1967.

[47] Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[48] Ronald R. Yager. A new approach to the summarization of data. *Information Sciences*, 28(1):69 – 86, 1982.

[49] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, pages 103–114. ACM, 1996.

[50] Chin-Yew Lin and Eduard Hovy. Identifying topics by position. In *Fifth Conference on Applied Natural Language Processing*, pages 283–290, 1997.

[51] Chinatsu Aone, Mary Ellen Okurowski, and James Gorlinsky. Trainable, scalable summarization using robust NLP and machine learning. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 62–66. Association for Computational Linguistics, 1998.

[52] Jaime G Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, volume 98, pages 335–336, 1998.

[53] Regina Barzilay, Kathleen R McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 550–557, 1999.

[54] Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. *Advances in automatic text summarization*, pages 111–121, 1999.

[55] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. *Advances in Automatic Summarization*, pages 55–60, 1999.

[56] Chin-Yew Lin. Training a selection function for extraction. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 55–62. ACM, 1999.

[57] Markus M Breunig, Hans-Peter Kriegel, Peer Kröger, and Jörg Sander. Data bubbles: Quality preserving performance boosting for hierarchical clustering. In *ACM SIGMOD Record*, pages 79–90. ACM, 2001.

[58] Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Fast hierarchical clustering based on compressed data and optics. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 232–242. Springer, 2000.

[59] John M. Conroy and Dianne P. O'leary. Text summarization via hidden Markov models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 406–407, 2001.

[60] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

[61] Jianjun Zhou and Jörg Sander. Data bubbles for non-vector data: Speeding-up hierarchical clustering in arbitrary metric spaces. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 452–463. VLDB Endowment, 2003.

[62] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the workshop on text summarization branches out (WAS 2004)*, pages 25–26, 2004.

[63] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[64] DK Evans, K McKeown, and JL Klavans. Similarity-based multilingual multi-document summarization. *IEEE Transactions on Information Theory*, 49, 2005.

[65] Varun Chandola and Vipin Kumar. Summarization – compressing data into an informative representation. *Knowledge and Information Systems*, 12(3):355–378, Aug 2007.

[66] Dipanjan Das and André F. T. Martins. A survey on automatic text summarization. Technical report, Carnegie Mellon University, 2007.

[67] Karen Spärck Jones. Automatic summarising: The state of the art. *Information Processing & Management*, 43:1449–1481, 2007.

[68] Oguzhan Tas and Farzad Kiyani. A Survey Automatic Text Summarization. *PressAcademia Procedia*, 5(1):205–213, 2007.

[69] Krysta Svore, Lucy Vanderwende, and Christopher Burges. Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 448–457, 2007.

[70] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[71] Giuliano Armano, Alessandro Giuliani, Alberto Messina, Maurizio Montagnuolo, and Eloisa Vargiu. Experimenting Text Summarization on Multimodal Aggregation. In *DART@ AI* IA*, 2011.

[72] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[73] Albaraa Abuobieda, Naomie Salim, Ameer Tawfik Albaham, Ahmed Hamza Osman, and Yogan Jaya Kumar. Text summarization features selection method using pseudo genetic-based model. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 193–197. IEEE, 2012.

[74] Elena Lloret and Manuel Palomar. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41, 2012.

[75] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer, 2012.

[76] Miguel Almeida and Andre Martins. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 196–206, 2013.

[77] Sanghoon Lee, Saeid Belkasim, and Yanqing Zhang. Multi-document text summarization using topic model and fuzzy logic. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 159–168. Springer, 2013.

[78] Fabricio E da S Tosta, Ariani Di Felippo, and Thiago AS Pardo. Estudo de métodos clássicos de sumarização no cenário multidocumento multilíngue. In *STUDENT WORKSHOP ON INFORMATION AND HUMAN LANGUAGE TECHNOLOGY (TILiC)*, volume 3, pages 1–3, 2013.

[79] Sherif Elfayoumy and Jenny Thoppil. A survey of unstructured text summarization techniques, 2014.

[80] Demetris Hoplaros, Zahir Tari, and Ibrahim Khalil. Data summarization for network traffic monitoring. *Journal of network and computer applications*, 37:194–205, 2014.

[81] Evangelos E Papalexakis, Alex Beutel, and Peter Steenkiste. Network anomaly detection using co-clustering. *Encyclopedia of Social Network Analysis and Mining*, pages 1054–1068, 2014.

[82] Mohiuddin Ahmed, Abdun Naser Mahmood, and Michael J. Maher. An Efficient Approach for Complex Data Summarization Using Multiview Clustering. In Jason J. Jung, Costin Badica, and Attila Kiss, editors, *Scalable Information Systems*, pages 38–47. Springer International Publishing, 2015.

[83] Mohiuddin Ahmed, Abdun Naser Mahmood, and Michael J. Maher. A Novel Approach for Network Traffic Summarization. In Jason J. Jung, Costin Badica, and Attila Kiss, editors, *Scalable Information Systems*, pages 51–60, Cham, 2015. Springer International Publishing.

[84] Ramakrishna Bairi, Rishabh Iyer, Ganesh Ramakrishnan, and Jeff Bilmes. Summarization of multi-document topic hierarchies using submodular mixtures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 553–563, 2015.

[85] Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. Multi-document abstractive summarization using ilp based multi-sentence compression. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[86] ZR Hesabi, Zahir Tari, A Goscinski, Adil Fahad, Ibrahim Khalil, and Carlos Queiroz. Data summarization techniques for big data—a survey. In *Handbook on Data Centers*, pages 1109–1152. Springer, 2015.

[87] Baotian Hu, Qingcai Chen, and Fangze Zhu. Lcsts: A large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865*, 2015.

[88] Nicoló Rivetti, Yann Busnel, and Achour Mostefaoui. Efficiently summarizing data streams over sliding windows. In *2015 IEEE 14th International Symposium on Network Computing and Applications*, pages 151–158. IEEE, 2015.

[89] D. K. Kanitha and D. Muhammad Noorul Mubarak. An Overview of Extractive Based Automatic Text Summarization Systems. *International Journal of Computer Science and Information Technology*, 8(5):33–44, 2016.

[90] Zubair Shah, Abdun Naser Mahmood, and Michael Barlow. Computing hierarchical summary of the data streams. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 168–179. Springer, 2016.

[91] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66, 2017.

[92] Hilário Oliveira, Rafael Dueire Lins, Rinaldo Lima, and Fred Freitas. A regression-based approach using integer linear programming for single-document summarization. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 270–277. IEEE, 2017.

[93] Rakesh Verma and Daniel Lee. Extractive summarization: Limits, compression, generalized model and heuristics. *Computación y Sistemas*, 21(4):787–798, 2017.

[94] Asad Abdi, Siti Mariyam Shamsuddin, and Ramiz M Aliguliyev. QMOS: Query-based multi-documents opinion-oriented summarization. *Information Processing & Management*, 54(2):318–338, 2018.

[95] Yue Dong. A survey on neural network-based summarization methods. *arXiv preprint arXiv:1804.04589*, 2018.

[96] Mohammed Javed, P Nagabhushan, and Bidyut B Chaudhuri. A review on document image analysis techniques directly in the compressed domain. *Artificial Intelligence Review*, 50(4):539–568, 2018.

[97] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. *ACM Computing Surveys (CSUR)*, 51(3):62, 2018.

[98] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K Reddy. Neural abstractive text summarization with sequence-to-sequence models. *arXiv preprint arXiv:1812.02303*, 2018.

[99] Eder Vázquez, Rene Arnulfo Garcia-Hernandez, and Yulia Ledeneva. Sentence features relevance for extractive text summarization using genetic algorithms. *Journal of Intelligent & Fuzzy Systems*, 35(1):353–365, 2018.

[100] Mohiuddin Ahmed. Data summarization: a survey. *Knowledge and Information Systems*, 58(2):249–273, 2019.

[101] Wasifa Chowdhury. *Employing neural hierarchical model with pointer generator networks for abstractive text summarization*. PhD thesis, Simon Frazer University: School of Computing Science, 2019.

[102] Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International journal of information management*, 35(2):137–144, 2015.

[103] Yin-Fu Huang and Yi-Hao Li. Sentiment translation model for expressing positive sentimental statements. In *2019 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pages 79–84. IEEE, 2019. DOI: `https://soi.org/10.1109/iCMLDE49015.2019.00025`.

[104] DS Maylawati, YJ Kumar, FB Kasmin, and MA Ramdhani. An idea based on sequential pattern mining and deep learning for text summarization. In *Journal of Physics: Conference Series*, page 077013. IOP Publishing, 2019.

[105] N. Kannaiya Raja, Naol Bakala, and S. Suresh. NLP: Text summarization by frequency and sentence position methods. *International Journal of Recent Technology and Engineering*, 8(3):3869–3872, 2019.

[106] MPJ van der Loo and E De Jonge. Data Validation Infrastructure for R. *Journal of Statistical Software (accepted for publication)*, 2019.

[107] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2021. Acessed: `https://github.com/probml/pml-book/releases/latest/download/book1.pdf`. ISBN: 0262046822.

[108] Chahira Touati and Amina Kemmar. Deep reinforcement learning approach for ontology matching problem. *International Journal of Data Science and Analytics*, pages 1–16, 2023.

[109] Marc Ehrig. *Ontology Alignment: Bridging the Semantic Gap*. Springer, Germany, 1 edition, 2007.

[110] Mariano Rodriguez-Muro, Roman Kontchakov, and Michael Zakharyaschev. Ontology-based data access: Ontop of databases. In *International Semantic Web Conference*, pages 558–573. Springer, 2013.

[111] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: A survey. In *International Joint Conferences on Artificial Intelligence*, 2018.

[112] Guohui Xiao, Davide Lanti, Roman Kontchakov, Sarah Komla-Ebri, Elem Güzel-Kalaycı, Linfang Ding, Julien Corman, Benjamin Cogrel, Diego Calvanese, and Elena Botoeva. The virtual knowledge graph system ontop. In *International Semantic Web Conference*, pages 259–277. Springer, 2020.

[113] Guohui Xiao, Linfang Ding, Benjamin Cogrel, and Diego Calvanese. Virtual knowledge graphs: An overview of systems and use cases. *Data Intelligence*, 1(3):201–223, 2019.

[114] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246, 2002.

[115] Jeffrey D Ullman. Information integration using logical views. In *International Conference on Database Theory*, pages 19–40. Springer, 1997.

[116] Alon Y Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, 2001.

[117] George Fletcher, Jan Hidders, Josep Lluís Larriba-Pey, et al. *Graph Data Management*. Springer, 2018.

[118] Alex Borgida and Ronald J Brachman. Conceptual modeling with description logics. In *The description logic handbook: theory, implementation, and applications*, pages 349–372. Cambridge University Press, 2003.

[119] Alex T Borgida, Vinay Chaudhri, Paolo Giorgini, and Eric Yu. *Conceptual modeling: foundations and applications: Essays in honor of John Mylopoulos*, volume 5600. Springer Science & Business Media, 2009. ISBN: 3642024629.

[120] The_W3C_SPARQL_Working_Group. SPARQL 1.1 Overview, 2013. `http://www.w3.org/TR/rdf-sparql-query/`. Acessado em 18/04/2021.

[121] Ivan Ricarte. Programação para a web semântica. Technical report, Unicamp, 11 2019.

[122] Jean-Baptiste Lamy. *Ontologies with Python*. Apress, 01 2021. ISBN: 978-1-4842-6551-2. DOI `https://doi.org/10.1007/978-1-4842-6552-9`.

[123] Jean-Baptiste Lamy. Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial intelligence in medicine*, 80:11–28, 2017.

[124] Marcelo Ferreira and Marcos Lopes. Linguistica Computacional. In Jose Luiz Fiorin, editor, *Novos caminhos da Linguistica*, pages 195–214. Editora Contexto, São Paulo, SP Brazil, 2017.

[125] Bhargav Srinivasa-Desikan. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd, 2018. ISBN: 978-1788838535123.

[126] Ferdinand de Saussure. *Curso de Linguística Geral*. Cultrix, 2012.

[127] Mark T. Keane Michael W. Eysenck. *Manual de Psicologia Cognitiva*. Artmed, 7 edition, 2017. ISBN: 978-85-8271-395-2.

[128] Dipanjan Sarkar. *Text analytics with Python*. Apress, 2016.

[129] Akshay Kulkarni and Adarsha Shivananda. *Natural language processing recipes: Unlocking text data with machine learning and deep learning using python*. Apress, 2019.

[130] Taweh Beysolow II. *Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*. Apress, 2018.

[131] Marcelo Ferreira and Marcos Lopes. *Para Conhecer Linguistica Computacional*. Editora Contexto, Sao Paulo, Brazil, 1 edition, 2020.