# ESTRUTURA PARA AQUISIÇÃO, REPRESENTAÇÃO E USO DE CONHECIMENTO, APRENDIZAGEM E COLABORAÇÃO ENTRE AGENTES E HUMANOS EM DOMÍNIOS DA INFRAESTRUTURA DA INTERNET

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**

O foco deste projeto de pesquisa é consolidar algumas atividades associadas a componentes do ambiente denominado Estrutura de Aquisição de Conhecimento, Uso, Aprendizagem e Colaboração (SKAU), proposto em tese de doutorado [1, 2]. O objetivo principal é popular uma Base de Conhecimento (KB) com conhecimento proveniente do repositório de documentos denominados *Request For Comments* (RFCs) do RFC Editor. Adicionalmente, o projeto pretende realizar a construção de ontologias de domínios com conhecimento retirado de ambientes afins, como por exemplo, o eco-sistema da Infraestrutura da Internet e domínios de conhecimento próximos. Três ferramentas são usadas para representar o conhecimento a ser capturado: Protégé 5, OnTop e a estrutura de dados de dicionário, das linguagens de programação. A base de conhecimento hospedeira do conhecimento capturado será usada por agentes autônomos presentes e organizados em uma estrutura denominada Arquitetura Autônoma sobre Domínios Restritos (A2RD), que colaboram entre si e entre seres humanos através de uma estrutura de listas inversamente encadeadas denominada IIBlockchain. As questões que serão resolvidas através de estudos primários e experimentações, se referem à captura e representação de conhecimento a partir de outras bases não estruturadas e da representação do conhecimento propiciados pelas Linguagens de Descrição e aperfeiçoadas pelo processo de alinhamento dos diversos resultados obtidos.

***Keywords*** ontologia · infraestrutura da Internet · base de conhecimento · ietf · irtf

## 1 Enunciado do Problema

A Internet é uma rede de redes de computadores. Tais redes de computadores são denominadas de Sistemas Autônomos (ASes) e possuem uma identificação única, como se pode ver pela Figura 1. Os ASes estão interligados por meio de uma complexa infraestrutura de telecomunicações, espalhada por toda a terra e além. A operacionalização dos ASes que usam esta infraestrutura de telecomunicações é feita por programas, isto é, algoritmos denominados genericamente de protocolos.

Este projeto está orientado a estudar e pesquisar técnicas e recursos para adequar um ambiente de suporte a agentes inteligentes com ações sobre domínios dos Sistemas Autônomos (ASes) ou Domínios de Roteamento [3].

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

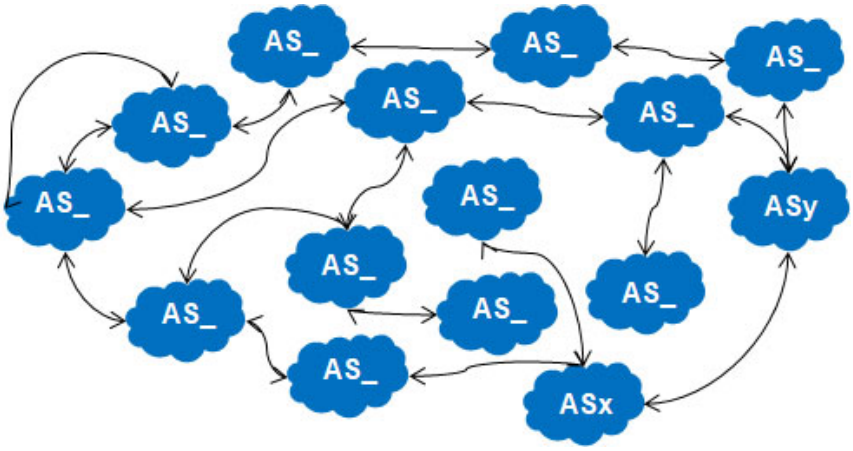


Figure 1: As redes autônomas de que constituem a Internet.

Os agentes, que também são algoritmos estão organizados em um modelo de camadas denominado Arquitetura Autônoma sobre Domínios Restritos (A2RD), dependem de uma base de conhecimento (KB) e de um ambiente de colaboração (IIBlockchain), componentes que pertencem a um eco-sistema denominado Estrutura de Aquisição de Conhecimento, Uso, Aprendizagem e Colaboração (SKAU) [1], cujo cenário pode ser visto na Figura 2.

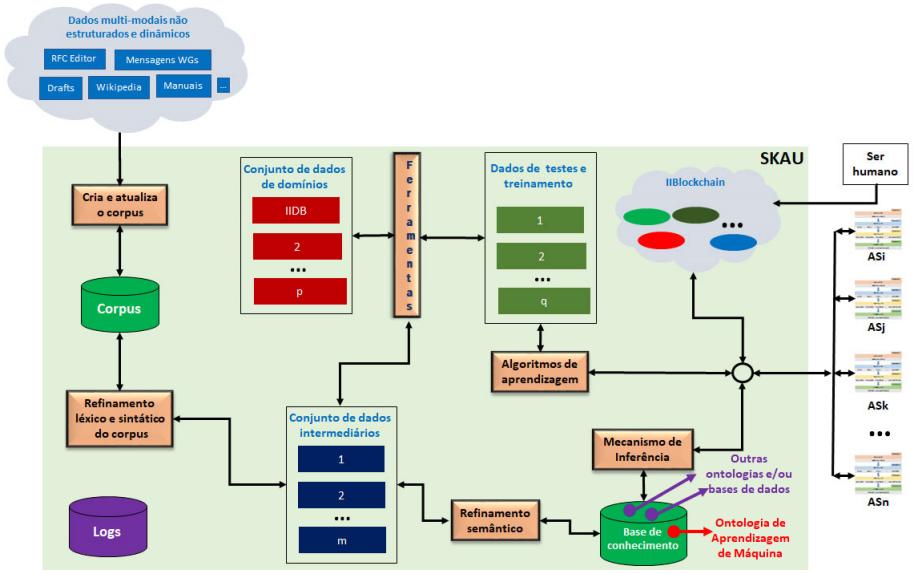


Figure 2: O ambiente SKAU, suporte para agentes do modelo A2RD, onde a Base de Conhecimento é a principal preocupação desta pesquisa. Adaptado de [1].

O foco principal desta pesquisa é a experiência no preenchimento da Base de Conhecimento (KB) com conhecimento obtido: (i) de bases de dados não estruturadas; (ii) da construção manual e/ou automática de ontologias de domínio. A pesquisa se baseia na aplicação sobre a Infraestrutura da Internet das técnicas e recursos de Inteligência Artificial, particularmente aquelas presentes na área de Aprendizagem de Máquina (ML), com maior intensidade, no uso das facilidades disponíveis no Processamento de Linguagem Natural (NLP), cuja relação está visível na Figura 3 [4] [5] [6] [7] [8] [9].

O preenchimento da KB envolve questões de pesquisas relacionadas com a representação de conhecimento capturado a partir fontes (estruturadas e não-estruturadas) externas e com as técnicas para o armazenamento deste conhecimento capturado.

Complementarmente, os agentes do modelo A2RD devem aperfeiçoar sua autonomia através de cooperação entre si e entre seres humanos. O mecanismo de colaboração utilizado pelos agentes é construído sobre uma base de dados de

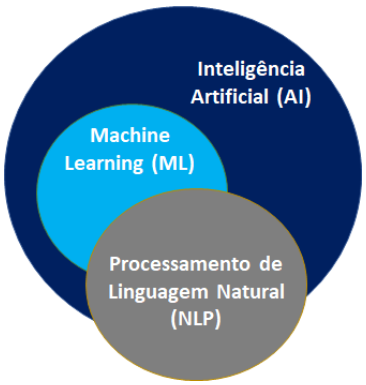


Figure 3: A relação entre AI, ML e NLP. Adaptado de [9].

listas encadeadas invertidas, denominada IIBlockchain [10]. O ferramental de uso da IIBlockchain está pronto e será intensivamente utilizado, quando apropriado, durante todo o período de duração da presente pesquisa. A IIBlockchain foi desenvolvida em Python, que, também será a principal linguagem de desenvolvimento do projeto.

Além desta Introdução, nomeada como Enunciado do Problema (Seção 1), este texto expõe a Metodologia que envolve o projeto, na Seção 3, com detalhes das técnicas, recursos e facilidades utilizadas. A Seção 4 indica os resultados esperados pelo presente projeto de pesquisa, no período programado, enquanto que a Seção 5 exibe os desafios a serem enfrentados diante do atual estágio da tecnologia e das técnicas envolvidas. A Seção 6 esclarece de que forma os resultados, as novas propostas e as lições aprendidas associadas aos esforços empreendidos no projeto serão disseminados e avaliados.

## 2 O ambiente da pesquisa

Os principais resultados esperados para este projeto de pesquisa virão de duas atividades de **captura de conhecimento**, cada uma com múltiplos processos e vinculadas a domínios da Infraestrutura da Internet. Tal captura de conhecimento ocorrerá a partir: de **bases não estruturadas** e, da **construção de ontologias de domínios**.

Todos os dados, algoritmos, programas e dados são de domínio público e estarão disponíveis no repositório do projeto [11]. Na sequência, o detalhamento de como o projeto irá considerar as atividades em cada uma das alternativas de captura e representação de conhecimento bem como as técnicas e recursos a serem utilizados [12].

### 2.1 Captura e representação de conhecimento a partir de bases não estruturadas

São muitas as bases não estruturadas originárias de conhecimento que interessam ao projeto, e entre elas estão aquelas produzidas pelas atividades do *Internet Engineering Task Force*[3] (IETF), do *Internet Research Task Force*[4] (IRTF) e de organizações associadas que orbitam em seu entorno formando um eco-sistema próprio de governança, parcialmente desenhado na Figura 4 [3].

A Figura 2 exibe, fora do ambiente SKAU, um conjunto de dados multi-modais não estruturados e dinâmicos, produzidos pelo IETF, IRTF e inúmeras outras instituições, governamentais, não governamentais e privadas, que fazem parte do eco-sistema caracterizado acima: RFC Editor, Mensagens geradas pelos Grupos de Trabalhos (WGs) do IETF e do IRTF, seus textos preliminares (*drafts*), textos da Wikipedia, manuais de produtos e facilidades, etc. Neste projeto, durante o primeiro ano somente o repositório do RFC Editor[5] será usado para formação da base de conhecimento, com a captura de conhecimento dos documentos disponíveis [13]. A Figura 5 ilustra os componentes envolvidos nesse processo.

Os documentos, armazenados no repositório do RFC Editor, não denominados *Request for Comments* (RFCs) e seu conteúdo representam os padrões de protocolos e os comportamentos técnicos propostos aos administradores dos Sistemas Autônomos (ASes), que permite a Internet funcionar cada vez melhor [3]. Os processos que tratam as RFCs podem ser simplificados como representado na Figura 6.

[3] `https://www.ietf.org`
[4] `https://www.irtf.org`
[5] https://www.rfc-editor.org/about/

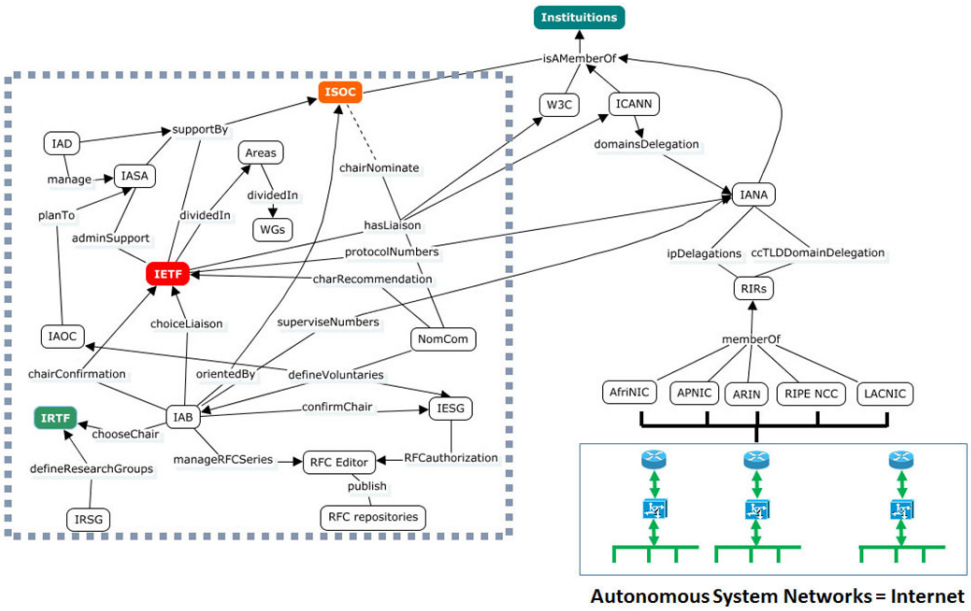


Figure 4: Eco-sistema da Infraestrutura da Internet com as instituições no entorno do IETF produzem documentos utilizados pelo SKAU para popular parcialmente a KB. Fonte: [1].

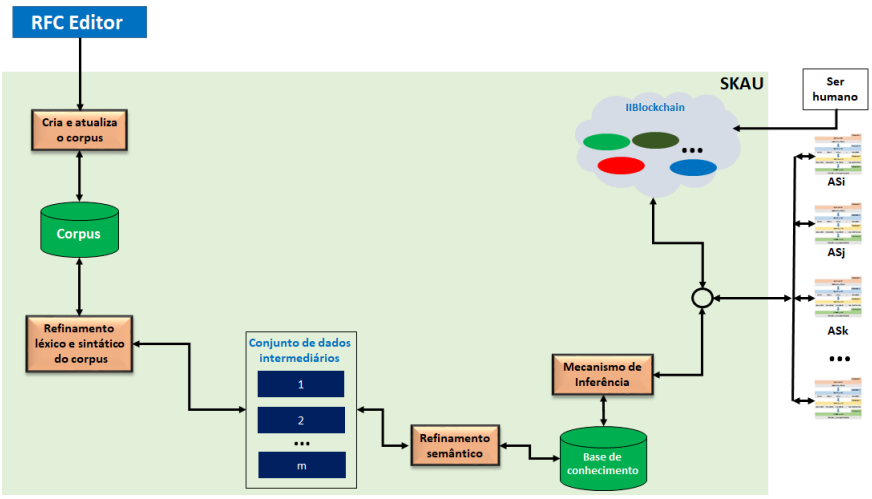


Figure 5: Interação com o repositório do *RFC Editor* para a captura de conhecimento e armazenamento na KB e principal foco da presente pesquisa. Adaptado de [1].

Embora o componente **Mecanismo de Inferência** permaneça na figura, ele somente será abordado durante as pesquisas associadas a ontologias. Trata-se de uma mecanismo que a partir do **Conhecimento Explícito** definidos pelas regras da ontologia, ele consegue obter o chamado **Conhecimento Implícito**.

O uso de um repositório "quase" estruturado como o das RFCs (a linguagem natural dos documentos é estruturada e alguns grupos de RFCs seguem um padrão) torna o uso da NLP mais fácil para programar e mais eficiente, pois pode-se usar uma técnica de criação de **corpus** em etapas denominado *pipeline*, de forma permanente (armazenada em meio digital) ou não, ilustrada pela Figura 7 mostrando as atividades desde a captura das RFCs, até a atualização da KB.

Um exemplo simples que justificaria a criação de um corpus é a captura de acrônimos e seu significado. Nas RFCs, um acrônimo e seu significado tem o seguinte formato padrão:

$$Significado\ do\ Acronimo \quad (ACRONIMO)$$

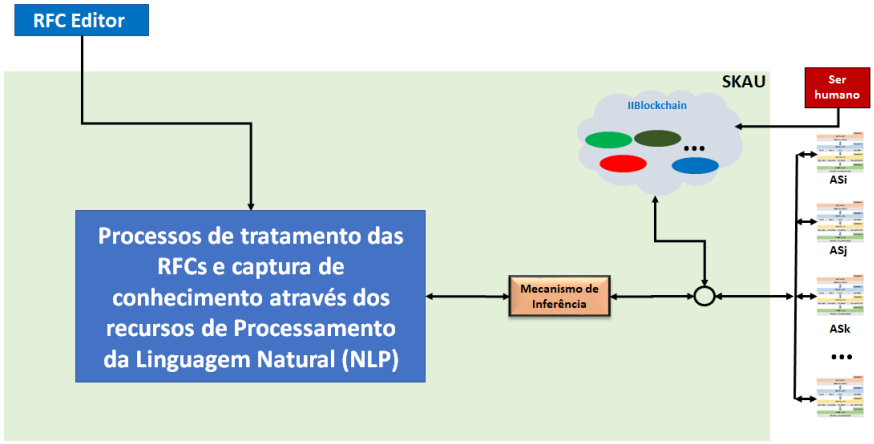


Figure 6: Núcleo do processamento das RFCs. Adaptado de [1].

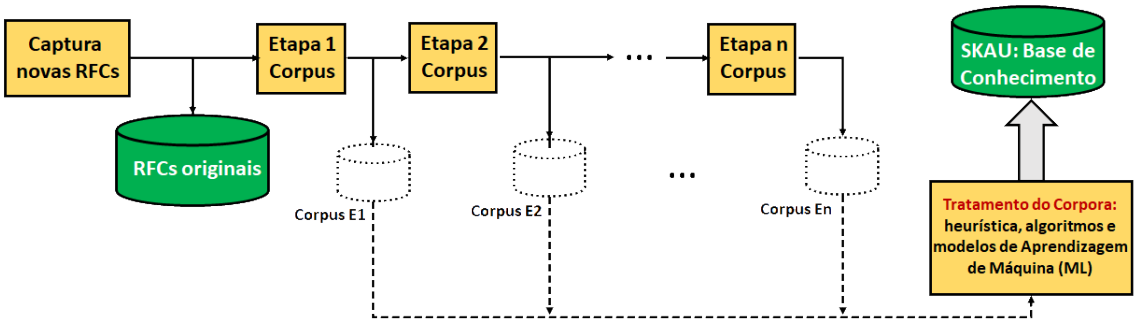


Figure 7: *Pipeline* de tratamento do corpora das RFCs

É intuitivo que o acrônimo e seu significado pertençam, completamente, a uma sentença. Então, a criação de um corpus no qual uma RFC esteja representada por suas sentenças distinguíveis, irá tornar o processamento da procura por acrônimo e seu significado bastante eficiente. Um algoritmo heurístico (usando pilhas) no qual procura-se por um "(" e um ")" dentro da sentença garante que o acrônimo é aquilo que está entre os parênteses.

Entretanto, alguns acrônimos são bem formados, como por exemplo:

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

outros não são tão imediatos assim:

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

A Figura 8 é a representação final de como serão tratadas as RFCs quando for usado os recursos agregados à NLP.

## 2.2 Captura e representação de conhecimento em ontologias de domínio

### 2.2.1 Preliminares

**Ontologia** foi definida em 1993 por Gruber e em 1997 foi melhorada mais apropriadamente por Borst [14] *apud* [15, 16]. Em 1998, Studer e outros ajustaram as duas definições na seguinte proposta: "Uma ontologia é uma especificação formal e explícita de uma conceituação compartilhada" [17] [14]. Vários autores refinaram as definições ao longo do tempo para indicar, mais claramente, que: **uma ontologia é um vocabulário formalmente definido para um domínio particular, utilizada para capturar conhecimento sobre esse domínio (restrito) de interesse**. Portanto, uma ontologia descreve os conceitos do domínio e também as relações que existem entre esses conceitos [18]. A inclusão de um domínio na definição facilitou a compreensão do significado da ontologia. Desta forma e finalmente,

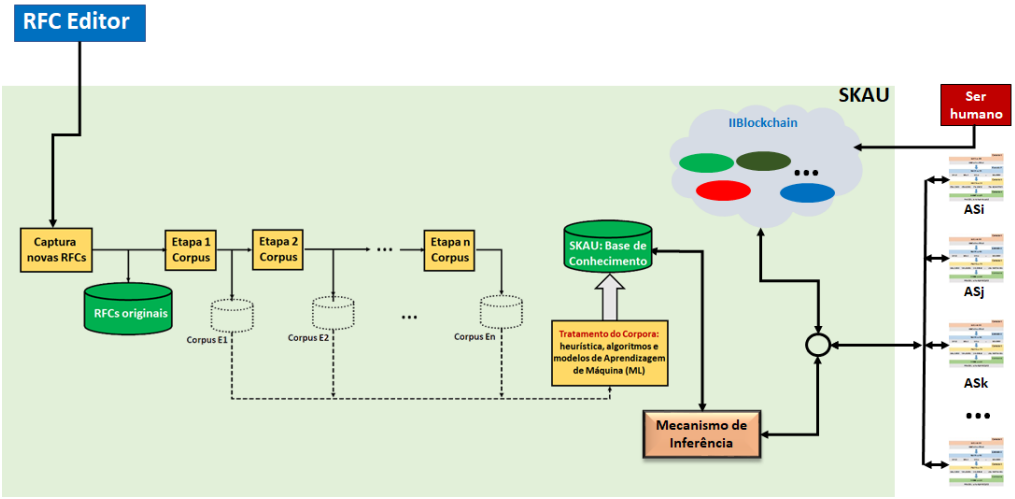


Figure 8: Componentes do SKAU diretamente envolvidos no presente projeto (em substituição à Figura 5), interação com o repositório do *RFC Editor* para a captura de conhecimento e armazenamento na KB, exibindo o tratamento das RFCs com o *pipeline* (Figura 7).

ontologias descrevem conceitos e relações em um domínio específico, para Representação do Conhecimento (KR) e Troca de Conhecimento (KE) [19].

Por outro lado, **base de conhecimento** (KB) é um conjunto de informações organizadas sobre um determinado assunto. Ela pode incluir fatos, regras, procedimentos e outros tipos de conhecimento. Uma base de conhecimento pode ser usada para apoiar uma variedade de aplicações, como sistemas de recomendação, sistemas de perguntas e respostas e sistemas de inteligência artificial.

A principal diferença entre ontologia e base de conhecimento é que a ontologia é um modelo formal de um domínio de conhecimento, enquanto a base de conhecimento é um conjunto de informações organizadas sobre um determinado assunto. A ontologia é mais abstrata e generalista, enquanto a base de conhecimento é mais concreta e específica.

Outra diferença importante é que a ontologia descreve os conceitos e as relações entre esses conceitos, enquanto a base de conhecimento pode incluir fatos, regras, procedimentos e outros tipos de conhecimento. A ontologia é, portanto, mais relevante para a representação do conhecimento estruturado, enquanto a base de conhecimento é mais relevante para a representação de conhecimento não estruturado.

A ontologia e a base de conhecimento podem ser usadas juntas para representar o conhecimento de um determinado domínio. A ontologia fornece a estrutura para a base de conhecimento, enquanto a base de conhecimento fornece os dados concretos que são usados para preencher a estrutura.

Aqui estão alguns exemplos de como ontologia e base de conhecimento podem ser usadas em conjunto:

- Um sistema de recomendação de produtos pode usar uma ontologia para representar os diferentes tipos de produtos disponíveis, como roupas, eletrônicos e alimentos. A base de conhecimento do sistema pode incluir informações sobre os produtos, como preços, avaliações e disponibilidade.
- Um sistema de perguntas e respostas pode usar uma ontologia para representar os diferentes tópicos sobre os quais ele pode responder. A base de conhecimento do sistema pode incluir informações sobre esses tópicos, como definições, exemplos e fatos.
- Um sistema de inteligência artificial pode usar uma ontologia para representar o conhecimento sobre o mundo real. A base de conhecimento do sistema pode incluir informações sobre o mundo real, como a localização de objetos, as relações entre objetos e os eventos que ocorrem no mundo.

### 2.2.2 Linguagens de representação do conhecimento

Uma das preocupações mais importantes na construção de uma ontologia é a forma como o conhecimento será representado. Em outras palavras, tal preocupação tem a ver com o formalismo (ou linguagem) usado para representar o conhecimento. Preferivelmente este formalismo deve expressar o conhecimento de um domínio, o mais completo

possível, sem que haja a possibilidade de ocorrer inconsistências, principalmente quando se fizer inferência sobre a KB, da qual a ontologia fará parte.

Em alguns casos, sempre que for necessário expressar conhecimento usando ferramentas gráficas ou outras representações menos extensas, mas com objetivo de esclarecer um conhecimento formal, o projeto usará os conceitos em torno das Lógicas de Descrição ou do inglês, *Description Logics* (DLs), que representam uma família de linguagens de representações de conhecimento, que podem ser usadas para representar (ou descrever) o conhecimento inserido em um domínio de aplicação, sob uma forma estruturada e bem caracterizada [20] [21]. Ao longo dos anos, muitas linguagens foram propostas e pesquisadas, principalmente sob precisão semântica de seus procedimentos de raciocínio [22]. No início surgiu a KL-ONE [23] que provocou o surgimento da K-REP [24], da BACK [25] e da LOOM [26], entre outras. Todas levaram à $\mathcal{SROIQ}$ [27], a DL que deu origem à *Web Ontology Language* (OWL) 1 e, finalmente, à OWL 2. A Figura 9 mostra as DLs derivadas da KL-ONE. O gráfico é resultado de uma ontologia, ainda em desenvolvimento, das DLs criadas para aperfeiçoar a representação do conhecimento, em ontologias.

### 2.2.3 A linguagem do projeto

A OWL 2 [28, 29], faz parte da família de DLs. É a linguagem preferida na construção de ontologias, no presente projeto, bem como outras, desde que aderentes a ela, tais como a *Resource Description Framework*[6] (RDF) e a *RDF Schema*[7] (RDFs).

## 3 Metodologia

O projeto será desenvolvido em duas etapas: (a) Construção manual e semi-automática, quando necessário, de uma ontologia parcial no domínio da Infraestrutura da Internet e (b) Construção de uma ontologia, sobre o mesmo domínio, com base nas técnicas de aprendizagem profunda (*deep learning*).

A ontologia manual e/ou semi-automática será construída através do **Protégé**[8] em sua versão 5. Diversas técnicas e algoritmos poderão ser utilizados para tornar esta ontologia parcial um modelo que exiba uma experiência adequada para a segunda etapa. Em paralelo aos trabalhos da primeira etapa será desenvolvida uma revisão sistemática de literatura em busca das principais recomendações mais recentes sobre as técnicas e algoritmos (alguns descritos abaixo) de aprendizagem profunda disponíveis para serem utilizados na segunda etapa.

A ontologia baseada em aprendizagem profunda será construída utilizando uma combinação de técnicas de Processamento de Linguagem Natural (PLN) e aprendizagem de máquina. Inicialmente, os dados brutos serão coletados de várias fontes confiáveis relacionadas à Infraestrutura da Internet e muitas delas identificadas pelo exercício manual e semi-automático da primeira parte. Esses dados serão então pré-processados para remover ruídos e torná-los adequados para análise.

Em seguida, um modelo de aprendizagem profunda será treinado nesses dados pré-processados. O modelo será configurado para identificar e aprender os conceitos e relações mais importantes no domínio da Infraestrutura da Internet. O resultado será uma ontologia rica e detalhada que captura o conhecimento nesse domínio de uma maneira estruturada e acessível.

Finalmente, as duas ontologias - a manual e a baseada em aprendizagem profunda - serão integradas para formar uma ontologia completa. Esta ontologia combinada será então validada por especialistas no domínio para garantir sua precisão e utilidade.

### 3.1 Algumas técnicas e algoritmos de aprendizagem profunda

A escolha da técnica de aprendizagem profunda dependerá muito do levantamento sistemático da literatura e dos dados coletados e a serem processados. É possível que serão experimentadas várias técnicas entre as quais serão escolhidas aquelas que oferecem os melhores perspectivas de se ajustarem aos objetivos do projeto.

Existem várias técnicas de aprendizagem profunda que podem ser utilizadas para a construção da ontologia, dependendo dos dados específicos e das necessidades do projeto. Aqui estão algumas possibilidades:

1. **Redes Neurais Convolucionais (CNNs)**: São especialmente eficazes no processamento de dados visuais e podem ser utilizadas se os dados incluírem imagens ou diagramas relacionados à Infraestrutura da Internet.

[6] `https://www.w3.org/TR/rdf-concepts/`
[7] `https://www.w3.org/TR/rdf-schema/`
[8] https://protege.stanford.edu/

Sabe-se que são os casos das Requeste for Comments (RFCs), disponíveis no repositório do RFC Editor[9]. São exemplos deste algoritmos: LeNet[10], AlexNet[11], VGG[12] (*Visual Geometry Group*), GoogLeNet[13], e ResNet[14].

2. **Redes Neurais Recorrentes (RNNs)**: São úteis para lidar com dados sequenciais ou temporais. Se os dados a serem coletados tiverem uma componente temporal (por exemplo, séries temporais de tráfego de rede), as RNNs podem ser uma boa escolha. Enre outras: LSTM[15] (*Long Short-Term Memory*) e GRU[16] (*Gated Recurrent Unit*) são exemplos de RNNs.
3. **Transformers**: Estes são modelos de aprendizagem profunda que utilizam mecanismos de atenção para melhorar a eficácia no processamento de dados sequenciais. Eles têm sido amplamente utilizados em tarefas de Processamento de Linguagem Natural (PLN). BERT[17] (*Bidirectional Encoder Representations from Transformers*), GPT[18] (*Generative Pretrained Transformer*), e T5[19] (*Text-to-Text Transfer Transformer*) são exemplos de Transformers.
4. **Autoencoders**: São redes neurais utilizadas para aprender representações codificadas (ou compactas) dos dados de entrada. Eles podem ser úteis para aprender a estrutura latente dos dados que serão coletados. VAE[20] (*Variational Autoencoder*) e *Denoising Autoencoder*[21] são exemplos de Autoencoders.
5. **Redes Geradoras Adversariais (GANs)**: São pares de redes neurais que trabalham em conjunto para melhorar continuamente a qualidade de eventuais previsões. Embora sejam mais conhecidas pela sua capacidade de gerar imagens realistas, elas também podem ser utilizadas em outros tipos de dados. Exemplos de GANs: DCGAN[22] (Deep Convolutional GAN), CycleGAN[23] e StyleGAN[24].

## 4 Resultados Esperados do projeto

A Popular a base de conhecimento (KB) do SKAU com conhecimento capturado das RFCs;
B Popular a KB do SKAU com conhecimento obtido através da construção de ontologias de domínios que interessam aos Agentes do modelo A2RD;

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

[10] Geralmente referindo-se ao LeNet-5, é uma CNN simples que se destaca no processamento de imagens em larga escala e foi uma das primeiras redes neurais convolucionais a impulsionar o desenvolvimento do aprendizado profundo (1998).

[11] É uma arquitetura de CNN, que foi projetada por Alex Krizhevsky em colaboração com Ilya Sutskever e Geoffrey Hinton. Ela se destacou na competição *ImageNet Large Scale Visual Recognition Challenge* em 30 de setembro de 2012.

[12] É uma arquitetura padrão de CNN, com várias camadas, sendo conhecida por suas versões VGG-16 e VGG-19, que consistem em 16 e 19 camadas convolucionais, respectivamente.

[13] Também conhecido como Inception V1, é uma arquitetura de CNN proposta pela Google em 2014, que utiliza módulos de *Inception* para escolher entre vários tamanhos de filtros convolucionais em cada bloco, permitindo a criação de arquiteturas mais profundas

[14] É um modelo de aprendizado profundo que utiliza conexões de identidade (ou "conexões residuais") para permitir o treinamento eficaz de redes profundas com dezenas ou centenas de camadas, melhorando a precisão ao aumentar a profundidade.

[15] É uma rede neural recorrente (RNN) que pode processar dados sequencialmente e manter seu estado oculto ao longo do tempo, sendo projetada para lidar com o problema do desaparecimento do gradiente presente nas RNNs tradicionais.

[16] É uma variante da rede neural recorrente (RNN) que utiliza portões de atualização e redefinição para resolver o problema do desaparecimento do gradiente, tornando-a mais eficaz no processamento de sequências longas.

[17] É um modelo de aprendizagem profunda que melhora a eficiência do processamento de linguagem natural (NLP), sendo famoso por sua capacidade de considerar o contexto ao analisar as relações entre as palavras em uma frase de maneira bidirecional.

[18] É um modelo de aprendizagem profunda que utiliza transformadores para gerar representações matemáticas complexas de texto ou outros tipos de mídia, permitindo que um computador execute algumas tarefas de maneira semelhante ao cérebro humano.

[19] É um modelo de aprendizagem profunda que transforma todas as tarefas de linguagem em uma tarefa de texto para texto, permitindo que o modelo aprenda de maneira mais eficaz a partir de dados não rotulados.

[20] É uma arquitetura de rede neural que comprime a entrada em uma representação de espaço latente e, em seguida, reconstrói a saída a partir dessa representação, sendo eficaz para aprender representações latentes.

[21] É um tipo de rede neural artificial que aprende a codificar dados de maneira eficiente de forma não supervisionada, com o objetivo principal de aprender uma representação (codificação) para um conjunto de dados, introduzindo uma restrição de reconstrução, e tem a capacidade de reconstruir a entrada a partir de uma versão corrompida, aprendendo efetivamente a remover ruído dos dados.

[22] É uma arquitetura de rede adversarial generativa que utiliza convoluções com passo (strided convolutions) e convoluções fracionadas para substituir as camadas de pooling, além de usar a normalização em lote (batchnorm) tanto no gerador quanto no discriminador.

[23] É uma técnica que permite a tradução entre dois domínios diferentes sem a necessidade de dados emparelhados, utilizando consistência de ciclo e treinamento adversarial para aprender mapeamentos entre dois domínios usando dados não emparelhados.

[24] É uma rede adversária generativa (GAN) que utiliza uma arquitetura alternativa para gerar imagens sintéticas de alta qualidade, oferecendo controle sobre o estilo da imagem gerada em diferentes níveis de detalhe.

C Aumento da KB com ontologias de domínio construídas por terceiros e alinhadas qualitativamente com as já existentes;
D Propor padrões de desenvolvimento de RFCs que facilite o uso de técnicas e algoritmos de ML. Por exemplo, propor padrões para criação de acrônimos evitando ambiguidades que possam criar problemas para as técnicas e algoritmos de ML;
E Documentar e difundir o tratamento de repositórios associado ao conhecimento de um domínio restrito;
F Desenvolver um processo de coleta do conhecimento seja feito pelos agentes do modelo A2RD. Por exemplo, o repositório de RFCs poderia ser trabalhado pelos agentes sem a intervenção humana, Inicialmente, os agentes construiriam ontologias com conhecimento minerado das RFCs. Um bom exemplo, são os acrônimos. A partir da forma geral de um acrônimo nas RFCs, que seria descoberto pelos próprios agentes, uma ontologia de acrônimos seria construída e populada automaticamente na KB. O resultado esperado nesta direção de automação será intermediário, isto é, espera-se que a experiência identifique os caminhos para tornar os agentes autônomos na construção de ontologias.
G Estabelecer uma análise de eficiência quantitativa das ontologias criadas usando DLs, via o Protégé por exemplo, os resultados pelo uso do OnTop e o conhecimento armazenado usando Dicionários. Adicionalmente, serão incluídas análises qualitativas do uso destes recursos sob o ponto de vista do ser humano, como usuário. (**Será intensivamente usado, a estrutura de dados de dicionário e o uso de bases de dados armazenadas em JSON, para disponibilizar os principais resultados da pesquisa, para uso de terceiros.**)
H Com a atenção voltada para as técnicas de NLP e as especificidades linguísticas caracterizadas pelo repositório de corpus gerenciados pelo RFC Editor, pode-se estabelecer que facilidades poderão ser implementadas ou indicadas para otimizar o trabalho da produção de *drafts* e/ou RFCs, evitando as não muito amigáveis ferramentas hoje disponíveis [3]. Também, um mecanismo de busca em RFCs poderá ser implementado, de forma bastante eficiente, ao recomendar as próximas palavras em uma sequência dada ou mesmo uma sentença ou texto, além de facilitar o estudo de RFCs orientadas a um assunto específico [30] [31] [32].
I Resultados de exercícios sobre os **modelos de linguagem** (incluindo as variações de **N-gramas**[25] confrontados com medidas de avaliação de **perplexidade**[26]), no repositório do RFC Editor poderá trazer outros resultados ainda não pensados neste momento.

### 4.1 Disseminação e avaliação

Além de trabalhos que deverão ser submetidos a eventos associados ao tema serão realizados dois seminários promovidos pelas instituições envolvidas no projeto. O primeiro seminário terá como tema principal **a construção de ontologias manuais ou semi-automatizadas**. O segundo seminário terá como tema central **o uso de aprendizagem profunda na contração de ontologias em domínio da Infraestrutura da Internet**.

### 4.2 Cronograma

A Etapa 1 correspondem a duas atividades (A1.1 e A1.2) que serão executadas em paralelo e são, respectivamente: (A1.1) construção manual e/ou semi-automática de uma ontologia para a Infraestrutura da Internet e (A1.2) levantamento sistemático da literatura sobre aplicação de aprendizagem profunda em ontologia. A Etapa 2 somente uma atividade (A2.1), correspondente a aplicação da aprendizagem profunda na construção de uma ontologia.

Ainda na Etapa 1, dois artigos estão programados para serem entregues. O primeiro (E1.1) se refere à experiência da construção manual e/ou semi-automática da ontologia proposta e o segundo (E1.2) se refere ao resultado da atividade A1.2, isto é, o resultado da revisão sistemática de literatura.

A atividade A2.1 terá com entregáveis, dois artigos para serem disponibilizados a publicações (E2.1 e E2.2).

Em cada semestre do projeto, está previsto um relatório técnico sobre os resultados do semestre e as lições aprendidas até o momento de sua publicação (R1, R2, R3 e R4).

No final de cada ano, serão promovidos seminários (S1 e S2) de disseminação e avaliação do estado da arte das duas etapas do projeto.

Complementarmente ao projeto, um registro de todas as suas atividades estará documentado no repositório do projeto, hospedado no OSF (*Open System Framework*), através de documentos, em especial o SKAU-CODEX como registra a Figura 14 e demais descrições feitas na Seção 6 [33].

Na Tabela 1, o cronograma visual das atividades.

[25] Nome popular dado a Cadeia de Markov
[26] É uma medida que se usa para avaliar modelos de previsão. Quanto menor a perplexidade, melhor o modelo

Table 1: Cronograma de execução das atividades por semestre.

| **Atividades** | **1$^o$ Sem.** | **2$^o$ Sem.** | **3$^o$ Sem.** | **4$^o$ Sem.** |
|---|---|---|---|---|
| A1.1 (Ontologia manual) | | | | |
| A1.2 (Revisão sistemática) | | | | |
| A2.1 (Aprendizagem profunda) | | | | |
| E1.1 (Artigo: ontologia manual) | | | | |
| E1.2 (Artigo: revisão sistemática) | | | | |
| E2.1 (Artigo: aprendizagem profunda) | | | | |
| E2.2 (Artigo: aprendizagem profunda) | | | | |
| R1 (Relatório técnico) | | | | |
| R2 (Relatório técnico) | | | | |
| R3 (Relatório técnico) | | | | |
| R4 (Relatório técnico) | | | | |
| S1 (1o. Seminário disseminação e avaliação) | | | | |
| S2 (2o. Seminário disseminação e avaliação) | | | | |

## 5 Desafios científicos e tecnológicos e as técnicas e métodos que podem ser usadas no projeto

Dois desafios tecnológicos devem ser considerados: (a) **representação do conhecimento** e a (b) **descoberta de conhecimento**.

Ambos são desafios complexos, mas o primeiro, é menos do que o segundo. Esta conclusão tira-se da literatura e da experiência do pesquisador em lidar com as duas questões. As linguagens de programação ao disporem de uma diversidade imensa de estrutura de dados, como o **dicionário**, por exemplo, ajudam a construir as propostas de **representação do conhecimento** que a ciência recomenda.

Mas, as recomendações da ciência para tratar computacionalmente representação do conhecimento, bem como a descoberta de conhecimento estão caracterizadas por paradigmas e propostas derivadas que tornam difícil a escolha.

As soluções para enfrentar estes dois desafios são a dedicação intensa aos estudos primários e à experiência (implementação das escolhas), sobre as quais se fala nas duas subseções que seguem.

### 5.1 Representação do Conhecimento

O desafio de representar o conhecimento possui um complemento que o antecede, conhecido como **captura** desse conhecimento. Na pesquisa que está sendo proposta, o domínio do conhecimento é estreito, quando se pensa no contexto do repositório do RFC Editor. Este fato facilita a atividade envolvida nesta questão e pode-se partir dos seguintes estudos primários para procurar e experimentar por uma alternativa técnica aceitável:

- [34]
- [35]
- [36]
- [37]
- [38]
- [12]
- [39]
- [40]
- [41]

### 5.2 Descoberta de Conhecimento

A este segundo desafio denomina-se **descoberta de conhecimento a partir do repositório das RFCs** (do inglês, *knowledge discovery from RFCs* ou KDR).

Os estudos primários relacionados com este desafio são muitos e os paradigmas estão bem diversificados. Apesar do domínio de conhecimento contido nas RFCs ser bastante estreito o esforço para uma escolha ou escolhas que desapareçam com o desafio depende de uma extensa disponibilidade de artigos acadêmicos. Desde o final de 2019 tem sido anotado no Codex do SKAU_UFABC uma série de informações a partir de um pesquisa não-sistemática da

literatura, a qual listou cerca de 62 (sessenta e dois) estudos primários a partir dos quais se poderia avaliar as alternativas, Tabela 2.

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

| # | Estudos Levantados | 2019 | 2021 |
|---|---|---|---|
| 1. | [42] | 3675 | 4095 |
| 2. | [43] | 26430 | 30470 |
| 3. | [44] | 1865 | 2038 |
| 4. | [45] | 420 | 456 |
| 5. | [46] | 5632 | 6237 |
| 6. | [47] | 379 | 404 |
| 7. | [48] | - | 77 |
| 8. | [49] | 2732 | 3099 |
| 9. | [50] | 463 | 509 |
| 10. | [51] | 1369 | 1479 |
| 11. | [52] | - | 1813 |
| 12. | [53] | 21 | 237 |
| 13. | [54] | 112 | 112 |
| 14. | [55] | 29 | 33 |
| 15. | [56] | 417 | 490 |
| 16. | [57] | 1813 | 601 |
| 17. | [58] | 35 | 36 |
| 18. | [59] | 3977 | 6093 |
| 19. | [60] | 1571 | 1818 |
| 20. | [61] | 4 | 45 |
| 21. | [62] | - | 166 |
| 22. | [63] RT | - | - |
| 23. | [64] | 395 | 444 |
| 24. | [65] | 456 | 545 |
| 25. | [66] | 234 | 277 |
| 26. | [67] | - | 629 |
| 27. | [68] | 3 | 4 |
| 28. | [69] | - | 570 |
| 29. | [70] | 40 | 51 |
| 30. | [71] | - | 239 |
| 31. | [72] | - | 501 |
| 32. | [73] | 76 | 87 |
| 33. | [74] | 14 | 16 |
| 34. | [75] | 2 | 3 |
| 35. | [76] | - | 12 |
| 36. | [77] | - | - |
| 37. | [78] | - | - |
| 38. | [79] | - | - |
| 39. | [80] | - | - |
| 40. | [81] | 10 | 22 |
| 41. | [82] | 85 | 119 |
| 42. | [83] | 15 | 25 |
| 43. | [84] | 40 | 193 |
| 44. | [85] | 5 | 23 |
| 45. | [86] | 1 | - |
| 46. | [87] | 4 | 6 |
| 47. | [88] | - | 393 |
| 48. | [89] | - | 9 |

continua na próxima página

**Table 2 – continuação da página anterior**

| # | Estudos Levantados | 2019 | 2021 |
|---|---|---|---|
| 49. | [90] | - | 20 |
| 50. | [91] | - | 31 |
| 51. | [92] | - | 17 |
| 52. | [93] | - | 17 |
| 53. | [94] | - | 134 |
| 54. | [95] | - | 50 |
| 55. | [96] | - | 17 |
| 56. | [97] | - | 24 |
| 57. | [98] | 1 | - |
| 58. | [99] | 2167 | 3377 |
| 59. | [100] | - | 1 |
| 60. | [101] | - | 1 |
| 61. | [102] | - | - |
| 62. | [103] | 1 | 5 |

Detalhes de cada artigo são resumidos no referido Codex. Estas anotações serão disponíveis no repositório do projeto.

As duas colunas da Tabela 2 representam as citações recebidas por cada artigo no final do ano de 2019, quando do levantamento e no momento de 2021, no qual o projeto foi apresentado. A partir de citações dos estudos primários é possível chegar a estudos mais recentes. Esta procura deve ser extensiva e critérios serão adotados (e registrados), para que o tempo seja suficiente para as conclusões necessárias.

Para os desafios descritos anteriormente deve-se ter em mente que a maior parte dos experimentos estará focada nos algoritmos de ML [7] [8] [104]. Se, face ao curto espaço de tempo, não for possível implementar toda a experimentação desejada, detalhada documentação a respeito estará exposta no Codex SKAU_UFABC para, eventualmente, serem implementados em oportunidades futuras.

### 5.3 Alinhamento de Ontologias

As atividades de pesquisas que envolvem ontologias estão associadas, principalmente, ao problema de alinhamento de ontologias. Refere-se à transformação de duas ou mais ontologias em uma única ontologia, onde as semelhanças são ajustadas adequadamente [105]. Uma visão intuitiva pode ser vista na Figura 10.

Os experimentos com o alinhamento de ontologias irão ampliar a perspectiva de popular a base de conhecimento, fortalecendo assim, a expectativa de conhecimento disponível para os agentes do modelo A2RD. Isto está exposto na Figura 11.

### 5.4 OBDA

Ontop é um sistema que implementa o OBDA, com arquitetura semelhante à mostrada na Figura 12 para construir grafos de conhecimento virtuais [107, 108, 109],

Ao invés de estruturar a camada de integração como uma coleção de tabelas relacionais, o ONTOP é estruturado como um Grafo de Conhecimento Virtual (do inglês, *Virtual Knowledge Graph* (VKG) [110]. Segundo este mesmo autor, a abordagem do VKG combina três ideias, trazidas abaixo via uma tradução livre:

- VKG.1 Virtualização de dados (do inglês, *data virtualization* (DV) [111] [112] [113], citados por [110] obtida evitando expor os utilizadores finais às fontes de dados reais, e apresentando-lhes, em vez disso, uma representação conceptual do domínio de interesse, tipicamente chamado esquema global.
- VKG.2 Os dados são estruturados sob a forma de um grafo (G), onde os objetos de domínio e os valores dos dados são representados como nós. As propriedades dos objetos são codificadas como arestas [114].
- VKG.3 O gráfico que representa os dados é enriquecido pelo conhecimento do domínio (KD), captando, por exemplo, hierarquias de conceitos e propriedades, domínio e gama de propriedades, e propriedades obrigatórias [115] [116]. Tal conhecimento permite inferir sobre os dados e conhecimentos, e assim derivar um novo conhecimento implícito do explicitamente afirmado.

O ONTOP faz uso extensivo da SPARQL [117] e possui uma integração com o Protégé.

### 5.5 Apache Jena

A implementação do Apache Jena[27] permitirá complementar a experimentação que pesquisas com a SPARQL[28] e ARQ[29], participar da representação em grafos, do RDF[30]. O Apache Jena usa SQL, permite integração com HTTP (Fuseki) e suporta o uso de OWL para construir ontologias.

A base de desenvolvimento do Jena é Java. Jena será uma das últimas ferramentas a serem analisadas, somente se, nenhuma das outras, não satisfazerem a pesquisa.

### 5.6 Ontologia em Python

Como Python é a linguagem de programação escolhida para o projeto, as bibliotecas disponíveis serão exploradas sempre que necessário e possível. Python, como uma linguagem dinâmica tem oferecido recursos e facilidades para tratar ontologias. A **RDFLib** oferece diversos métodos para extrair informações de grafos RDF [118]. Mais recentemente a biblioteca **Owlready** está se mostrando apropriada para tratar ontologias com muita facilidade via Python, incluindo base de dados, à semelhança de Ontop [119] [120].

Outras facilidades da linguagem Python em ontologias serão descobertas e documentadas no repositório do projeto.

### 5.7 Linguística Computacional

Linguística Computacional é um campo multidisciplinar que usa técnicas de Processamento de Linguagem Natural (NLP) envolvendo a Inteligência Artificial, principalmente sua sub-área Aprendizagem de Máquina (ML), aplicadas à área de Linguística [121].

Os corpus mantidos pelo RFC Editor fazem parte de um domínio bastante apropriado e restrito, para experimentações produzidas por **modelos de linguagem**, por demais técnicas e recomendações alicerçadas na linguística computacional [122], na linguística geral [123] e em particular, na ciência cognitiva computacional [124].

São muitas as técnicas de NLP que serão exploradas e documentadas detalhadamente neste projeto de pesquisa. Há bastante referências disponíveis e relacionadas com NLP usando Python, abordando principalmente a estrutura linguística do inglês [125] [126] [127].

### 5.8 Processamento da Linguagem Natural

O processamento do córpora gerenciado pelo RFC Editor (com mais de 9.000 documentos) pode exigir uma esforço computacional muito grande e uma capacidade de memória bastante alta. Por exemplo, nos modelos de Markov (n-gramas) o tamanho cresce exponencialmente. Supondo um vocabulário $V$: um modelo bigrama o tamanho será de $V^2$; um modelo trigrama, $V^3$; um modelo quadrigrama, $V^4$; e, assim sucessivamente [128].

Tais desafios poderão ser enfrentados com mudanças nas demandas de memória dos algoritmos aumentando o custo de processamento, que demandaria poder computacional. Alternativas de uso de recursos computacionais poderosos disponíveis no Brasil, como por exemplo, o Sistema de Computação Petaflópica do SINAPAD[31].

No momento apropriado as alternativas serão avaliadas, apontando a solução adequada e disponível.

### 5.9 Normalização e Alinhamento das Ontologias produzidas pelas diversas ferramentas

Este é um desafio que aparece quando do uso de ferramentas para produzir ontologias. Os resultados podem ser diferentes, inclusive em um mesmo domínio. Um processo de normalização e alinhamento adaptam os resultados antes de seguirem para popular o contêiner KB. Admite-se que, eventualmente, seja possível o uso de outras ferramentas, ainda desconhecidas. A Figura 13 ilustra este processo.

[27] `https://jena.apache.org`
[28] `https://jena.apache.org/tutorials/sparql_pt.html`
[29] `https://jena.apache.org/documentation/query/manipulating_sparql_using_arq_pt.html`
[30] `https://jena.apache.org/tutorials/rdf_api_pt.html`
[31] `https://lncc.br/sinapad/`

## 6 Registro das lições aprendidas durante o desenvolvimento do projeto

As lições aprendidas durante o período de pesquisa e desenvolvimento serão registradas em um conjunto de documentos que farão parte do chamado SKAU_CODEX, onde um extrato pode ser visto na Figura 14. Oportunamente, este material será compilado em textos de divulgação tipo tutoriais para que sejam aproveitados por estudantes e pesquisadores das respectivas áreas. Esses documentos poderão ser publicados no blogue Infraestrutura da Internet[32], em forma de *preprints*, no Wiki do repositório do projeto SKAU-UFABC [33], no repositório do projeto (SKAU Project Repository) Os membros do projeto estarão sempre disponíveis para disponibilizar palestras, entrevistas ou outra forma de divulgação explicitamente relacionada como projeto.

Finalmente, informações sobre o andamento do projeto e seus principais resultados estarão sistematicamente disponíveis no Twitter, LinKedin e Instagram. Eventualmente, outros canais da mídia digital poderão ser usados.

## 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.

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

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

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

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

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

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


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

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

[18] 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.

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

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

[21] 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.

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

[23] 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.

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

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

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

[27] 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.

[28] 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/.

[29] 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.

[30] 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.

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

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

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

[34] 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.

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

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

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

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

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

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

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

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

[43] 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.

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

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

[46] 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.

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

[48] 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.

[49] 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.

[50] 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.

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

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

[53] 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.

[54] 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.

[55] 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.

[56] 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.

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

[58] 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.

[59] 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.

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

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

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

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

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

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

[66] 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.

[67] 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.

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

[69] 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.

[70] 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.

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

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

[73] 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.

[74] 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.

[75] 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.

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

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

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

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

[85] 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.

[86] 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.

[87] 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.

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

[89] 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.

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

[91] 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.

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

[93] 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.

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

[95] 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.

[96] 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.

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

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

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

[100] 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`.

[101] 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.

[102] 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.

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

[104] 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.

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

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

[107] 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.

[108] 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.

[109] 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.

[110] 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.

[111] 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.

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

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

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

[115] 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.

[116] 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.

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

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

[119] 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`.

[120] 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.

[121] 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.

[122] 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.

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

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

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

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

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

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

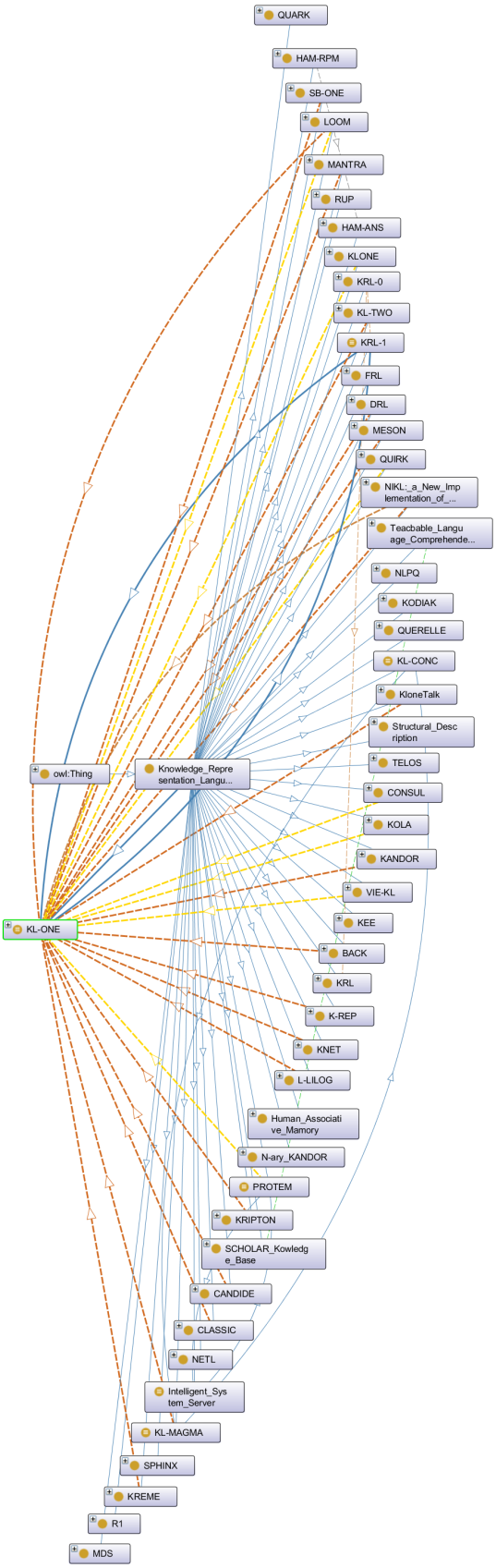


Figure 9: Ilustração parcial de Linguagens de Representação de Conhecimento onde a maioria delas são derivadas da KL-ONE (representado separadamente) estabelecida em 1978. Gráfico obtido através do *plugin* **OntoGraf**, do Protégé.

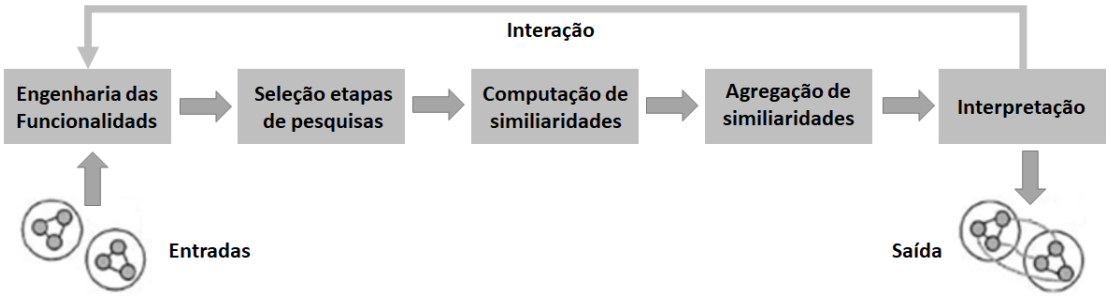


Figure 10: A entrada, saída, e as seis principais etapas envolvidas no processo de alinhamento. Adaptado de [106]

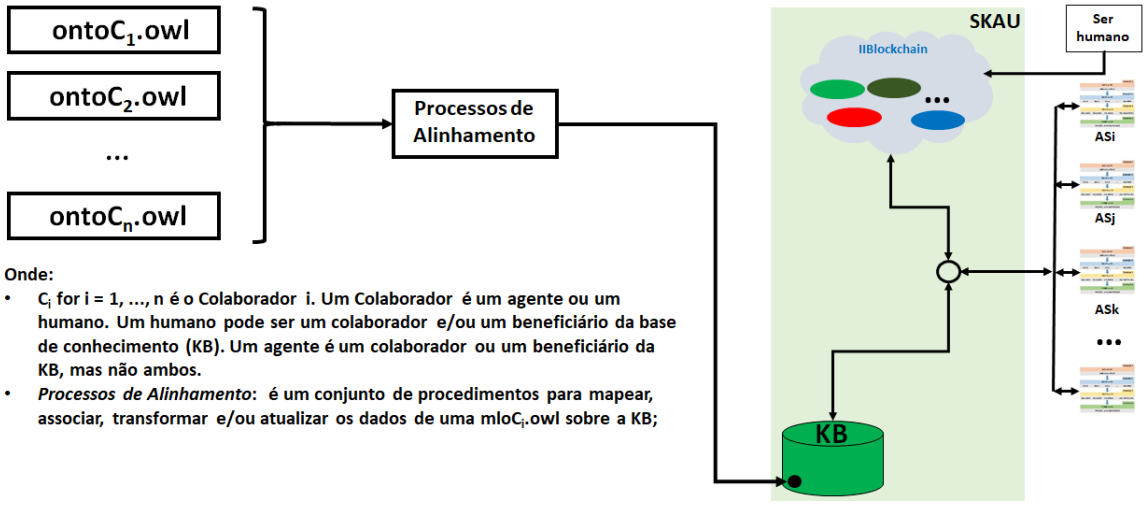


Figure 11: Como as ontologias de domínio são inseridas na base de conhecimento através de processos de alinhamento

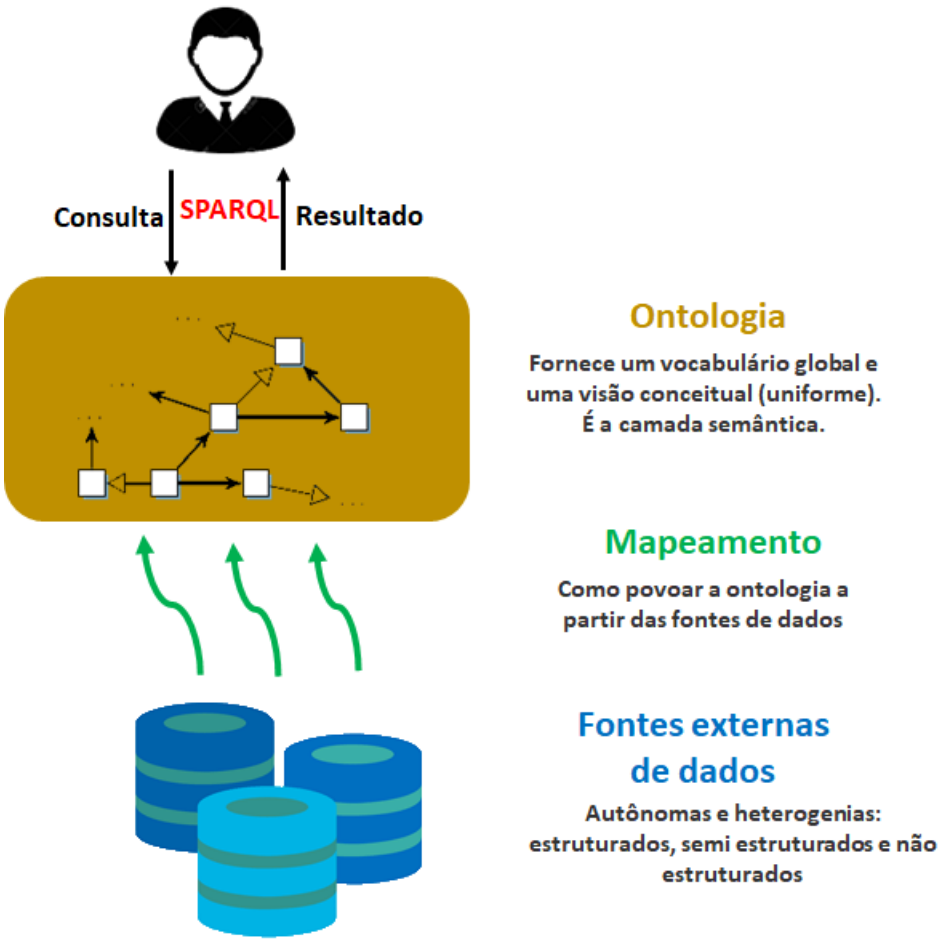


Figure 12: O que é o OBDA

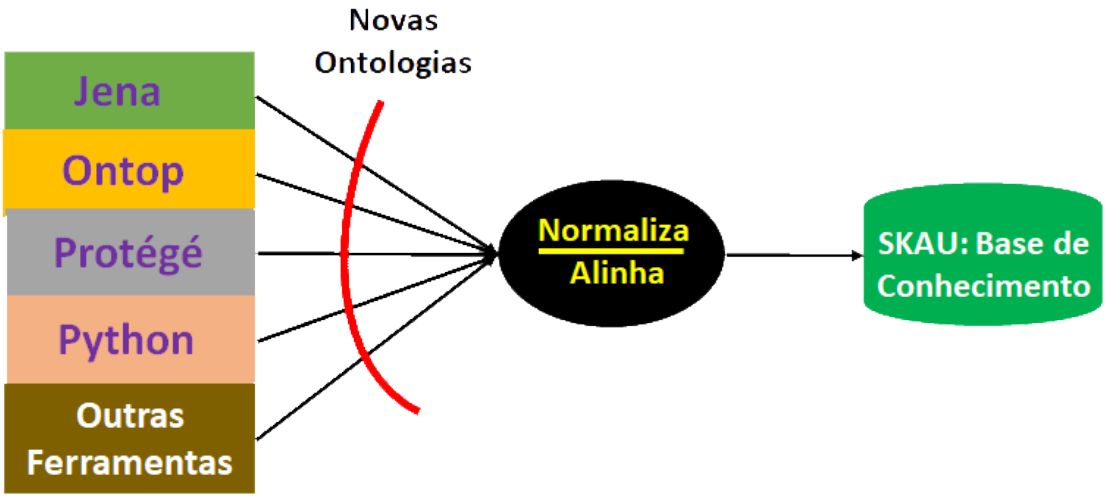


Figure 13: Como os diversos resultados de ontologias produzidas por diferentes ferramentas são tratados antes de armazenadas na KB

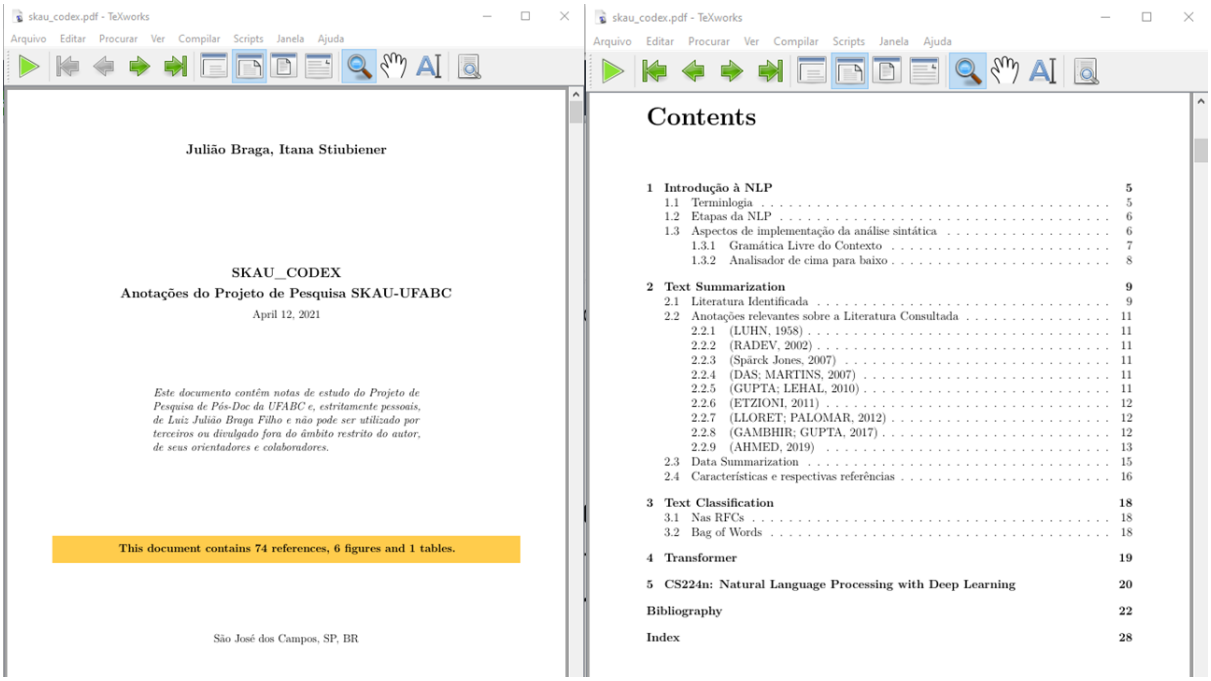


Figure 14: SKAU_CODEX: extrato do documento. Fonte: Autores