UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO
EM COMPUTAÇÃO APLICADA
NÍVEL DOUTORADO

DENIS ANDREI DE ARAÚJO

ENSEPRO:

Engenho Semântico de Pergunta e Resposta baseado em Ontologia

São Leopoldo
2019

Denis Andrei de Araújo

ENSEPRO:
Engenho Semântico de Pergunta e Resposta baseado em Ontologia

Tese apresentada como requisito parcial para a obtenção do título de Doutor, pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – UNISINOS

Orientador: Dr. Sandro José Rigo

São Leopoldo

2019

A663e Araújo, Denis Andrei de.
Ensepro : engenho semântico de pergunta e resposta baseado em ontologia / Denis Andrei de Araújo. – 2019.
[147] f. : il. ; 30 cm.

Tese (doutorado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2019.
"Orientador: Dr. Sandro José Rigo."

1. Sistema de pergunta e resposta semântico. 2. Processamento da linguagem natural (computação). 3. Grafos. 4. Word embedding. I. Título.

CDU 004

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecária: Amanda Schuster – CRB 10/2517)

Chegar ao fim desta caminhada somente tornou-se possível com o apoio e incentivo de algumas pessoas, às quais dedico este trabalho. Primeiramente minha família, em especial minha amada esposa Carolina Müller e meus queridos filhos Daniel e Mariana. No âmbito profissional, contei com o constante apoio e a revigorante compreensão dos estimados colegas César Habitzreuter e André Nunes. Por fim, dedico esta vitória à minha mãe, Dona Ana, e meu pai, Seu Darcy, pedras fundamentais e alicerces de toda e qualquer conquista por mim alcançada.

AGRADECIMENTOS

Inicio agradecendo o constante e fundamental apoio recebido do Prof. Dr. Sandro Rigo, que mais do que orientar o trabalho, associou-se a esta empreitada, desde o início até ao fim da caminhada. Foi por seu intermédio que tive o privilégio de trabalhar com os bolsistas João Henrique Muniz e Alencar Rodrigo Hentges, a quem devo agradecer a dedicação e parceria na realização das atividades decorrentes diretamente da pesquisa. Agradeço também o suporte e atenção recebida da secretaria do Programa de Pós-Graduação em Computação Aplicada (PPGCA), na pessoa da srta. Luciana Grimaldi.

Agradeço também à CAPES pelo apoio financeiro para a consecução não somente do doutorado em si, mas também para a viabilização do estágio realizado na Universidade de Évora em Portugal, estágio este realizado sob a orientação do Prof. Dr. Paulo Quaresma, a quem agradeço o acolhimento e apoio.

Por fim, agradeço à compreensão e apoio da empresa em que trabalho, a CWI Software, nas pessoas dos colegas César Habitzreuter e André Nunes, os quais empenharam-se pessoalmente na busca do apoio oficial da empresa no cumprimento das atividades relacionadas ao doutorado.

RESUMO

Há uma grande expectativa em relação ao uso da linguagem natural como interface de comunicação com as máquinas. Dentre as várias aplicações que implementam tal interface, despontam os sistemas de Pergunta e Resposta Semânticos, os quais possibilitam a localização de informações em bases de conhecimento a partir de perguntas formuladas em linguagem natural. Percebe-se nos trabalhos em andamento uma tendência à implementação de soluções baseadas nas informações léxicas e morfológicas das perguntas, desprezando-se as informações abstratas de nível mais elevado do processamento linguístico. Esta tese apresenta uma abordagem que explora de forma aprofundada as informações sintáticas e estruturais das perguntas, fundamentando-se nestes níveis mais elevados da linguística para depreender o significado de frases e assim localizar respostas nas bases de conhecimentos semânticas. A abordagem propõem um modelo que faz uso das informações linguísticas da pergunta para determinar o seu tipo e selecionar as palavras chaves que serão utilizadas para a busca de respostas na base de conhecimento. Ao contrário de outros trabalhos, o modelo propõem uma solução baseada em linguística integrada a outras duas diferentes técnicas de implementação, visando apresentar uma solução que explore as vantagens que cada técnica oferece. A abordagem usa as informações morfossintáticas e estruturais da frase para determinar o tipo da pergunta e para selecionar as suas palavras chaves. Posteriormente, utiliza novamente as informações linguísticas para otimizar o desempenho do algoritmo baseado em grafo para geração e ranqueamento de candidatas a resposta. Por fim, caso o uso integrado das informações linguísticas com a técnica baseada em grafos não seja suficiente para a seleção inequívoca das respostas, a abordagem busca apoio na semântica latente do word embedding para validar as respostas. Os experimentos de avaliação da abordagem mostraram um desempenho acima dos demais concorrentes, apresentando Escore F1 micro de 0,56 e Escore F1 macro de 0,593.

Palavras-Chave: Sistema de Pergunta e Resposta Semântico. Processamento da Linguagem Natural. Grafos. Word Embedding.

ABSTRACT

There is great expectation regarding the use of natural language as an interface of communication with machines. Among the several applications that implement such an interface, the Semantic Question Answering systems arises, enabling the localization of information in knowledge bases from questions formulated in natural language. It is possible to notice in the work in progress a tendency to implement solutions based on the lexical and morphological information of the questions, ignoring the higher level abstract information of the linguistic processing. This thesis presents an approach that explores in depth the syntactic and structural information of the questions, based on these higher levels of linguistics to understand the meaning of the words and to find answers in semantic knowledge bases. This approach proposes a model that makes use of the linguistic information of the question to determine its type and select the keywords that will be used to search answers in the knowledge base. Unlike other works, the model proposes a solution based on linguistics integrated with two different implementation techniques, aiming to present a solution that exploits the advantages that each technique offers. The approach uses the morphosyntactic and structural informations of the sentence to determine the type of the question and to select its key words. Later, it uses linguistic information to optimize the performance of the algorithm of generation and ranking of candidates for the response based on graph. Finally, if the integrated use of linguistic information with the graph-based technique is not enough for the unequivocal selection of the answer, our approach look for support in the latent semantics of word embedding to validate the answers. The experiments of evaluation of the approach showed a performance above the other competitors, with a score F1 micro of 0.56 and F1 score Macro of 0.593.

Keywords: Semantic Question Answering. Natural Language Processing. Graphs. Word Embedding.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Os sistemas de Pergunta e Resposta (PR) têm como função a geração de respostas a perguntas formuladas em linguagem natural. Estes sistemas provêm uma interface mais natural entre homem e máquina, normalmente via texto ou fala, e tem por objetivo localizar respostas para o maior número possível de questões (SAWANT; CHAKRABARTI; RAMAKRISHNAN, 2018).

Os sistemas de PR visam responder questões objetivas relacionadas a fatos, de forma a ser possível julgar o quão correta é a resposta em relação a um padrão preestabelecido, possibilitando assim avaliar-se a precisão e a revocação do mecanismo de localização de respostas. Outra característica importante em relação as capacidades dos sistemas de PR diz respeito à complexidade do que se está perguntando. Uma pergunta pode ser formulada contendo várias questões combinadas, pressupondo então várias respostas. Estas perguntas mais longas demandam um esforço computacional maior por parte do sistema de PR no momento de processar a Linguagem Natural (HERMJAKOB, 2001).

Assim, além de definir se o sistema de PR tem como objetivo responder a perguntas sobre fatos, usualmente se estabelece também se haverá suporte a perguntas longas ou curtas. Perguntas curtas geralmente são expressas através de sentenças únicas com resposta também única, como os exemplos apresentados no Quadro 1.

***Quadro 1: Estrutura do índice utilizado para armazenar a BC no Elasticsearch*.**

1. Quando foi a Batalha de Gettysburg?
2. Liste todos os musicais com músicas de Elton John.
3. Qual a altura do Farol em Colombo?
4. Quem foi a esposa do presidente americano Lincoln?
5. Quanto custou Pulp Fiction?

Em relação às respostas, os sistemas de PR podem ser projetados para responder perguntas de uma área específica, quando então são chamados de sistema de PR de domínio fechado. Já os sistemas de PR que não impõem restrições em relação à área de conhecimento são ditos de domínio aberto.

A fonte de informação que o sistema de PR vai utilizar para localizar a resposta pode ser um conjunto de documentos contendo texto ou bancos de dados. Quando o sistema de PR é implementando visando a localização de respostas em Bases de Conhecimento semânticas (BC), costuma-se dizer que ele é um sistema de Pergunta e Resposta Semântico (DIEFENBACH et al., 2018a).

Esta tese tem por objetivo apresentar uma abordagem para o desenvolvimento de um sistema de Pergunta e Resposta Semântico (PRS) de domínio aberto para perguntas curtas. Propõe-se neste trabalho uma abordagem inovadora baseada principalmente na exploração das características linguísticas como insumo fundamental para a realização dos processos necessários para a compreensão da Linguagem Natural (LN) e a localização da resposta na BC, mas também apoiada nas principais técnicas atualmente utilizadas para localização e seleção das respostas. Propõe-se uma abordagem em que, além do tradicional uso de etiquetas de POS, busque-se a compreensão da LN utilizando-se também as estruturas linguísticas mais

complexas, combinando esta abordagem fundamentada na exploração profunda das informações linguísticas com o uso de subgrafos para localização e geração de candidatas à resposta, por fim apoiando-se em técnicas de Aprendizagem de Máquina (AM) para a seleção e validação das respostas.

1.1 Motivação

Os sistemas de PRS aplicam as técnicas e ferramentas de Processamento de Linguagem Natural (PLN) da Inteligência Artificial (IA) visando dotar o sistema com um nível de compreensão da pergunta recebida de forma a ser possível a localização da resposta em uma BC semântica.

Embora capacitar os sistemas computacionais a compreender a LN seja um desafio que tem motivado pesquisas desde os primórdios da computação (BOBROW, 1964; SWANSON, 1960), esta é uma área de pesquisa relevante (CHEN; ZHANG; ZHANG, 2019; FUTRELL; GRUBER, 2019; HEBERT et al., 2019). A busca por uma interface mais natural para os sistemas computacionais é um campo ao mesmo tempo desafiador e instigante. Isto não é diferente no que se refere a sistemas de PRS (BURON et al., 2019; DIEFENBACH et al., 2018b; HU et al., 2018).

São muitas as possibilidades de se utilizar a LN em sistemas automatizados, como bem demonstra o relatório publicado pelo Gartner[1] apontando as plataformas conversacionais e os assistentes virtuais próximas ao topo das expectativas quanto às tecnologias emergentes. A quantidade de trabalhos desenvolvidos no meio acadêmico, especificamente no que concerne a sistemas de PR (ABDUL-KADER; WOODS, 2015; DIEFENBACH et al., 2018a; HÖFFNER et al., 2017), vem a confirmar as expectativas do uso da LN como base para a comunicação homem-máquina.

Neste sentido, devido ao desenvolvimento da web semântica (BERNERS-LEE et al., 2001) nas últimas duas décadas, tem-se disponibilizado uma quantidade cada vez maior de dados estruturados na forma de BC (DIEFENBACH et al., 2018a). Os sistemas de PRS são implementações de interfaces que visam automatizar o processo de localização de informações sobre informações representadas em BC semânticas a partir de perguntas expressas em LN.

São muitos os desafios para aqueles que tem como objetivo desenvolver um sistema de PRS, tais como: suporte a BC de múltiplos domínio, escalabilidade visando capacidade de trabalhar com BC extensas (com dezenas de milhões de fatos, como a DBPedia, por exemplo), extração de informações a partir de múltiplas BCs interligadas, capacidade de compreensão de perguntas formuladas em múltiplas linguagens, entre outras tantas questões.

Analisando-se as publicações de trabalhos dos últimos dez anos, percebe-se três tipos principais de abordagens para a implementação de sistemas de PRS: aquelas fundamentadas primordialmente em grafos, as baseadas em informações linguísticas da pergunta e as abordagens que fazem uso de Aprendizagem de Máquina.

[1] https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018

As abordagem baseadas em grafos (DIEFENBACH; SINGH; MARET, 2018; HUANG; ZOU, 2013; XU; FENG; ZHAO, 2014) geralmente usam as palavras que compõem a pergunta em LN para gerar subgrafos, os quais são utilizados para elaborar a consulta SPARQL que, ao ser executada, obtém como retorno as respostas para a questão recebida. Em geral, são utilizadas listas de palavras a serem descartadas (stop words) e ferramentas de PLN para remoção das variações terminológicas, como a lematização (JACQUEMIN; TZOUKERMANN, 1999). Tais abordagens têm como principal vantagem o fato de serem independentes de idioma, pois ao fazerem uso das informações morfológicas das palavras (ou seja, somente consideram as letras que as compõem), podem aplicar praticamente a mesma abordagem a vários idiomas. A principal desvantagem de utilizar-se somente as informações morfológicas recai sobre o problema de maior imprecisão deste nível linguístico (VIEIRA; LIMA, 2001). Ao desconsiderar as informações linguísticas dos níveis mais altos de abstração, como a sintática e semântica, por exemplo, é impossível às abordagens baseadas em grafos discernir qual o significado da palavra e, por consequência, definir a sua importância e relevância para a localização da resposta.

As abordagens baseadas em informações linguísticas (BAUDIŠ, 2015; DAMLJANOVIC; AGATONOVIC; CUNNINGHAM, 2011; GONDEK et al., 2012; HAMON; GRABAR; MOUGIN, 2017), por sua vez, buscam apoio em informações linguísticas com níveis de abstração mais altos para o processamento das palavras que compõem a pergunta, buscando um nível de compreensão mais profunda das palavras que compõem a questão. Em princípio, estas abordagens podem ter uma maior precisão para interpretar o significado das palavras, já que os níveis de abstração mais elevados permitem discernir melhor tanto a função quanto a importância da palavra no contexto da pergunta. Contudo, abordagens com ênfase nas informações linguísticas de nível mais elevado são mais dependentes da língua em que a questão foi formulada, uma vez que vão considerar as informações linguísticas de nível mais elevado (sintática, semântica e estrutural) ao buscarem a compreensão do que foi expresso na questão, o que é considerado como um ponto negativo da abordagem baseada em linguística, embora esta seja uma questão discutível. Um algoritmo que tenha como meta compreender a linguagem natural e ser independente de idioma remete, em princípio, a uma situação conflitante.

No entanto, percebe-se atualmente que há uma tendência recente na implementação de algoritmos que possibilitem o desenvolvimento de sistemas computacionais que permitam a Compreensão da Linguagem Natural de forma independente da linguagem (RADFORD et al., 2018; YOUNG et al., 2018), geralmente utilizando técnicas baseadas em inferência estatística ou probabilística como base para o PLN. Esta são as técnicas utilizadas pelos sistemas de PRS baseados em Aprendizagem de Máquina (AM). Os sistemas de PRS baseados nesta abordagem, fortemente baseada no nível linguístico léxico e morfológico, são geralmente independentes de idioma, mas apresentam como principal vantagem o fato de aprenderem com base em treinamento sobre dados. Este é, sem dúvida um diferencial bastante expressivo, pois sistemas baseados em AM tem a capacidade de adaptarem-se ao domínio de contexto através de treinamento supervisionado. Em relação às duas outras abordagens, a adaptação ao domínio por treinamento é um diferencial importante, principalmente em relação a abordagem linguística, geralmente baseada em sistema de regras estático, embora a precisão da abordagem linguística em relação à baseada em AM ainda seja um ponto forte, reforçado pelo fato de que sistemas baseados em regras dispensam dados para treinamento.

Observa-se na prática que os trabalhos apresentados geralmente optam por uma técnica em detrimento das demais (HÖFFNER et al., 2017) em vez de uma abordagem que tenha por objetivo combinar o que de melhor cada uma oferece. O desafio de identificar-se os pontos mais fortes e buscar-se uma arquitetura robusta e flexível que maximize o desempenho de cada abordagem projeta uma perspectiva desafiadora e motivante. É neste contexto de alta expectativa em relação às interfaces em LN e de mobilização da comunidade científica em relação ao desafio da interpretação da LN para a localização de informações em BC semânticas que propõe-se nesta tese o desenvolvimento de uma abordagem para implementação de um sistema de PRS que explore as características linguísticas da pergunta de forma a melhor compreender a linguagem natural, fazendo uso de grafos para localização e geração de triplas candidatas, que busque apoio em técnicas de AM para validar as respostas.

1.2 Questão de pesquisa

Com base no contexto apresentado, a questão de pesquisa deste trabalho pode ser resumida na seguinte pergunta: em que medida a combinação de diferentes abordagens fundamentadas no uso das informações linguísticas de nível de abstração mais elevado presentes em frases da linguagem natural podem ajudar um sistema de PRS no processo de compreensão de perguntas expressas em LN e localização da respectiva resposta?

Tendo em vista esta questão, tem-se por objetivo apresentar neste trabalho um modelo híbrido para o desenvolvimento de um sistema de PRS que localize respostas em BC semânticas para perguntas curtas formuladas em língua portuguesa que explore profundamente as informações linguísticas das perguntas para implementação dos algoritmos necessários ao funcionamento do sistema.

1.3 Justificativa

A abordagem proposta nesta tese parte da hipótese que a combinação das três principais abordagens de implementação é factível na implementação de sistemas de PRS, explorando as vantagens de cada uma das implementações, fazendo uso dos aspectos morfológicos, sintáticos e estruturais da linguagem natural para realizar um alinhamento dos termos relevantes da pergunta com os relacionamentos e conceitos representados na BC, usando a abordagem baseada em subgrafos para a geração combinatorial de candidatas e buscando auxílio nas técnicas de AM para seleção e validação das respostas.

Em relação ao uso combinado das três técnicas de implementação de sistemas de PRS, naqueles trabalhos que fazem uso das informações linguísticas como base para o funcionamento do sistema (BAUDIŠ, 2015; GONDEK et al., 2012; HAMON; GRABAR; MOUGIN, 2017), observou-se nos trabalhos analisados que não se verifica nestes uma exploração profunda das características linguísticas das questões. No que concerne às duas outras abordagens, percebe-se uma tendência à concentração na técnica principal em detrimento das demais, tanto nas abordagens baseadas em grafos (DIEFENBACH; SINGH; MARET, 2018; HUANG; ZOU, 2013; XU; FENG; ZHAO, 2014), mas especialmente nas implementações baseadas em AM (HE et al., 2014; SHEKARPOUR et al., 2015; WANG; LING; HU, 2019). Considera-se então o uso aprofundado das informações linguísticas

combinado à aplicação das outras duas técnicas como um diferencial da abordagem aqui proposta.

Destaca-se também, tanto como uma justificativa quanto um diferencial da pesquisa desenvolvida neste trabalho, o foco na língua portuguesa, uma vez que, tanto quanto levantado, são escassos os sistemas de PRS que tenham como objetivo o processamento de perguntas formuladas em português.

1.4 Metodologia

Em relação aos aspectos metodológicos considerados no desenvolvimento deste trabalho, primeiramente realizou-se uma pesquisa bibliográfica para coletar informações sobre o tema em análise. O segundo passo envolveu uma análise das implementações de sistemas de PRS para identificação de procedimentos de interesse do trabalho. No terceiro passo foi especificada a solução a ser desenvolvida, junto com a definição de tecnologias para o desenvolvimento de um protótipo e de aspectos de avaliação.

O trabalho desenvolvido caracteriza-se como uma pesquisa de caráter exploratório (PRODANOV; DE FREITAS, 2013), pois a mesma busca inicialmente descrever em detalhes um determinado cenário no qual se identifica o problema tratado. Com base neste ponto inicial, busca-se demonstrar possibilidades para o encaminhamento de uma solução, que neste sentido está sendo descrita e validada com base no estudo de um caso de aplicação.

A metodologia de trabalho adotada segue as definições de Wazlawick (2017): a) realização de revisão bibliográfica, estudo documental e de implementações de sistemas de Pergunta e Resposta baseados em ontologia; b) realização de estudo e análise de trabalhos relacionados com as áreas tratadas; c) Modelagem e uma solução com os dados obtidos nas etapas anteriores; d) Descrição de um protótipo e de uma abordagem para avaliação; e) realização de estudo de caso para avaliar os resultados possíveis; f) descrição e documentação da pesquisa.

1.5 Organização do trabalho

Esta tese inicia com uma introdução em que são apresentadas a motivação e justificativas, os objetivos a serem alcançados, a metodologia de trabalho bem como a questão de pesquisa. Apresenta-se então no capítulo dois um referencial teórico contendo alguns conceitos básicos visando o bom entendimento do trabalho desenvolvido.
Segue-se então no capítulo três fazendo-se uma apresentação sintética de trabalhos correlatos, acompanhados de uma análise crítica contextualizada em relação ao trabalho desenvolvido nesta tese. Apresenta-se no capítulo quatro a descrição detalhada do modelo, seguida no capítulo cinco pelo detalhamento da implementação do protótipo e respectiva avaliação de desempenho.
Por fim, fecha-se a tese com a apresentação das considerações finais, fazendo um apanhado geral do trabalho, apresentando-se as considerações finais contendo as contribuições científicas, os pontos fortes e fracos da abordagem proposta, bem como as possibilidades de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os principais conceitos relacionados ao foco da pesquisa proposta neste trabalho. Apresenta-se primeiramente uma visão geral dos sistemas de PR, em especial aqueles que se utilizam de tecnologias semânticas para o tratamento da linguagem natural.

Por fim, fecha-se o capítulo discorrendo-se sinteticamente sobre as quatro principais tecnologias da Web Semântica, pilares fundamentais de todo e qualquer sistema que processe informações com base em tecnologias semânticas.

2.1 Processamento da Linguagem Natural

A linguagem é uma forma de comunicação que é efetivada pela troca de mensagens representadas por uma combinação específica de sinais gráficos ou acústicos entre comunicantes que compartilham um senso comum de conhecimento. A linguagem possui níveis de análise: a pragmática, que engloba o meio em que vivem os participantes da comunicação; a semântica, que são as relações entre as expressões da linguagem e os seus significados; e, por fim, o nível de sintaxe que examina as propriedades e estruturas da linguagem. O léxico e a morfologia são subníveis do nível sintático e dizem respeito às palavras e sua formação, como as flexões, derivações e a sua composição (LIDDY et al., 2003).

Ambiguidades podem ser produzidas em cada um dos níveis da linguagem, as quais podem ser resolvidas pelos níveis subsequentes mais elevados de análise. No entanto, algumas ambiguidades somente podem ser resolvidas pelo conhecimento do contexto em que a frase está inserida. É senso comum que os seres humanos normalmente utilizam todos estes níveis de análise linguística para a desambiguação do significado.

O Processamento de Linguagem Natural (PLN) refere-se a abordagem computadorizada para o tratamento das ambiguidades utilizando uma combinação de níveis de análise linguística, baseada em um conjunto específico de teorias e tecnologias. São os diferentes conjuntos de níveis de análise linguística que cada sistema de PLN utiliza que os diferenciam entre si. Além do conjunto de níveis utilizados, os sistemas de PLN podem diferenciar-se também pelo tipo de análise que implementam: superficial ou aprofundada (BENVENISTE, 1964).

O PLN é considerado uma disciplina da Inteligência Artificial, uma vez que trata do processamento da linguagem humana. Geralmente utilizado como ferramenta de apoio para a execução de tarefas ou para o desenvolvimento de aplicações específicas, está dividido em duas áreas: processamento da linguagem e geração da linguagem. O processamento refere-se à análise da linguagem para a produção de uma representação do significado, enquanto que a geração trata da produção da linguagem a partir da representação. Devido ao escopo deste trabalho, será vista somente a análise da linguagem natural, uma vez que é esta a área relevante para a metodologia aqui sugerida (MANNING; MANNING; SCHÜTZE, 1999).

2.1.1 Os níveis de processamento da linguagem

Uma abordagem que permite explicar o que realmente acontece dentro de um sistema de PLN envolve analisar os níveis de análise da linguagem que o sistema implementa. Pesquisas na área da psicolinguística sugerem que o processamento da linguagem é altamente dinâmico, com níveis que podem interagir em diversos momentos. Por este motivo, apresenta-se uma descrição resumida dos principais níveis de análise da linguagem escrita (BENVENISTE, 1964).

2.1.1.1 Morfológico

Este nível analisa a formação das palavras, buscando a sua decomposição em morfemas – a menor unidade com significado (LIDDY et al., 2003). Por exemplo, a palavra "predestinado" pode ser morfologicamente separada em três morfemas: o prefixo "pre", o radical "destino" e o sufixo "ado". Uma das estratégias que o ser humano pode utilizar para descobrir o significado de uma palavra desconhecida é decompô-la em morfemas e, sendo estes conhecidos, concluir o significado da palavra. Esta estratégia pode ser utilizada também pelos sistemas de PLN, por exemplo ao verificar as desinências verbais e assim concluir o tempo em que ocorre a ação expressa pelo verbo.

2.1.1.2 Léxico

Neste nível, o significado é interpretado pela análise individual das palavras. Existem várias abordagens para se chegar a compreensão do significado no nível de palavras, sendo a técnica de associação de etiquetas part-of-speech (POS) às palavras a abordagem mais comumente utilizada (VIEIRA; LIMA, 2001). Outra técnica utilizada no nível lexical é substituir as palavras que tem somente uma interpretação possível por uma representação semântica do seu significado. A natureza da representação depende da teoria semântica adotada no sistema de PLN.

O nível lexical pode requerer o uso de um léxico para a interpretação do significado. A abordagem adotada na implementação do sistema de PLN determina se um léxico será ou não utilizado, bem como o tipo e a extensão das informações nele representadas. Os léxicos podem ser bem simples, contendo somente a palavra e as suas possíveis etiquetas POS, mas podem ser também extremamente complexos e conter informações sobre a classe semântica da palavra, argumentos e respectivas limitações semânticas e até mesmo o campo semântico de cada sentido de uma palavra polissêmica (ALVES, 2001; LIDDY et al., 2003).

2.1.1.3 Sintático

Este nível analisa as palavras de uma sentença visando descobrir a estrutura gramatical da frase. Este tipo de análise requer uma gramática e um analisador. A saída produzida por este nível de processamento é uma representação que revela os relacionamentos de dependência estrutural entre as palavras de uma sentença (COVINGTON, 1994). Estão disponíveis uma grande variedade de gramáticas, as quais determinam a escolha do analisador. A sintaxe é importante para o significado porque a ordem e a dependência entre as

palavras influenciam o significado da sentença. Por exemplo: as frases “O menino feriu o cachorro” e “O cachorro feriu o menino” têm diferenças somente à nível de sintaxe, mas seus significados são muito diferentes.

2.1.1.4 Semântico

O processamento semântico determina os possíveis significados de uma sentença pela análise das interações entre os significados das palavras que a compõe. Este nível do processamento pode incluir a desambiguação semântica de palavras com múltiplos significados, selecionando somente um sentido da palavra polissêmica para a representação semântica da sentença (VIEIRA; LIMA, 2001). Quando a informação que determina o significado de uma palavra origina-se dos demais componentes da sentença diz-se que a desambiguação ocorreu a nível semântico. Existem vários métodos para se conseguir a desambiguação em nível semântico, como por exemplo a utilização da frequência em que ocorre determinado significado num corpus específico, métodos que consideram o contexto local e aqueles que utilizam o conhecimento pragmático do domínio do documento.

2.1.1.5 Pragmático

Este nível se preocupa com o uso intencional da linguagem e utiliza o contexto em detrimento do conteúdo textual para alcançar o entendimento. Visa explicar como um significado implícito é identificado em textos onde este sentido não é referenciado explicitamente. Requer conhecimento profundo do mundo real, incluindo a compreensão de intenção, planejamento e objetivos (LIDDY et al., 2003). O alcance deste nível de análise para aplicações de PLN pode demandar o uso de bases de conhecimento e módulos de inferência. Um exemplo clássico de uso seria a aplicação da análise pragmática para a resolução da dubiedade em referências anafóricas ambíguas resolvíveis (BENVENISTE, 1964).

2.1.2 Categorias e aplicações da PLN

Existem quatro categorias principais de classificação para sistemas de PLN: simbólica, estatística, conexionista e híbrida (WERMTER; RILOFF; SCHELER, 1996). As abordagens simbólica e estatística são utilizadas desde os primórdios da PLN.

Embora a abordagem simbólica tenha predominado inicialmente, com o advento da grande disponibilização de recursos computacionais ocorrida na década de 80, a abordagem estatística ganhou popularidade, principalmente devido à necessidade de desempenho para lidar com os vastos bancos de dados do mundo real. Também nesta época, a abordagem conexionista reconquista seu espaço, demonstrando a utilidade da aplicação de redes neurais no PLN.

Qualquer aplicação que manipule textos pode fazer uso das teorias e implementações que o PLN provê para melhor alcançar seus objetivos. No campo da Recuperação da Informação, por exemplo, a intensa manipulação de documentos a caracteriza como uma forte candidata ao uso das técnicas de PLN. No entanto, o uso de PLN em sistemas comerciais via de regra restringem-se ao nível morfológico, o que limita consideravelmente a precisão destes sistemas.

Uma área de aplicação do PLN são os sistemas de Pergunta e Resposta, que visam localizar informações para responder questões elaboradas em linguagem natural. Estas respostas podem ser utilizadas por aplicações tais como sistemas de busca, sistemas de conversação, assistentes pessoais, chatbots, entre outros.

Embora o PLN seja uma área de pesquisa relativamente recente se comparada a outras abordagens da tecnologia da informação, relatos de sucesso do uso desta técnica sugerem que a tecnologia de acesso à informação baseada em PLN permanece sendo uma área interessante de pesquisa e desenvolvimento de sistemas de informação.

2.2 Sistemas de Perguntas e Respostas

Na Ciência da Computação a área de pesquisa dedicada especificamente ao desenvolvimento de sistemas de PR tem por objetivo responder questionamentos feitos por seres humanos em linguagem natural utilizando técnicas de Recuperação da Informação e de Processamento da Linguagem Natural (STUPINA et al., 2016).

As respostas são elaboradas através de consultas a bases de dados estruturados contendo conhecimento ou informação. Estas bases de dados são normalmente denominadas de Bases de Conhecimento (LEVY; RAJARAMAN; ORDILLE, 1996). Outra possibilidade é que os sistemas de PR formulem suas respostas a partir de dados não estruturados, tais como documentos contendo texto em linguagem natural (PAŞCA, 2003).

Os sistemas de PR podem ser utilizados com vistas a responder sobre os mais variados assuntos como, por exemplo, questionamentos sobre fatos ou definições, ou perguntas do tipo “como” ou “por que”, etc. Mas, a principal divisão em relação ao tipo do sistema de PR está relacionada com a sua capacidade de responder questões sobre um domínio fechado ou aberto (STRZALKOWSKI; HARABAGIU, 2006).

Os sistemas de domínio aberto tem por objetivo tratar questões sobre qualquer assunto, geralmente consultando informações representadas em BC muito extensas, mais genéricas e de conhecimento geral (YANG et al., 2003). Sistemas de PR para domínio fechado lidam com questionamentos sobre uma área de conhecimento específico, como medicina, mecânica, turismo, etc. Para responder os questionamentos, os sistemas de PR podem acessar BC específicos, frequentemente formalizadas em ontologias de domínio (MOLLÁ; VICEDO, 2007).

Os sistemas de PR para domínio fechado podem restringirem-se a responder perguntas de um tipo específico, simplificando assim o processo de interpretação da linguagem natural. Os assistentes pessoais com interface em linguagem natural para aplicações bem específicas, como agendas pessoais, previsão do tempo, são exemplos típicos deste tipo de sistema.

Além do escopo de domínio das perguntas, os sistemas de PR podem também ser classificados em relação a forma pela qual as respostas são geradas. Uma abordagem possível é utilizar dados estatísticos gerados a partir da análise de corpus combinados com técnicas de Aprendizagem de Máquina (AM) para a predição de resposta (VINYALS; LE, 2015).

As abordagens baseadas em AM tem como principal característica a forte dependência do tamanho do corpus usado para treinamento. O desempenho do sistema está diretamente relacionado à quantidade de frases utilizadas na etapa de treinamento do algoritmo. A

vantagem desta abordagem é que o sistema de PR tem uma menor dependência do esforço humano para obtenção de melhor desempenho.

Uma outra possibilidade para a geração das respostas é fazer uso de ferramentas de PLN para a geração de informações linguísticas da pergunta para então, a partir destas informações, classificar o tipo de pergunta e de resposta esperada, bem como a identificação de qual é a informação desejada pelo usuário (WANTROBA; ROMERO, 2014).

A vantagem da abordagem linguística é a não dependência de treinamento sobre corpus extensos para o seu funcionamento. Em contrapartida, o seu desempenho está diretamente ligado ao esforço humano de predição e implementação de regras que identifiquem e processem corretamente as frases digitadas pelo usuário.

A escolha por uma ou outra técnica para a implementação do sistema de PR vai depender principalmente do contexto de aplicação. Se houver disponibilidade de expressiva quantidade de exemplos de perguntas e respostas coletadas anteriormente, por exemplo, o uso de técnicas de AM é uma boa possibilidade. Quando, ao contrário, não houver disponibilidade expressiva de dados anteriores para treinamento, a abordagem baseada em regras torna-se a solução mais viável.

Uma outra situação em que sistemas de PR baseado em regras são mais indicados é quando a resposta para a pergunta encontra-se em bancos de dados estruturados. Nos casos em que a resposta depende de uma consulta a um banco de dados ou base de conhecimentos, a opção de basear-se somente em regras ou combinar as técnicas em um sistema híbrido, o qual utiliza a técnica de AM para identificação do tipo ou domínio da pergunta (XU; SARIKAYA, 2014).

No contexto da pesquisa proposta neste trabalho, o foco da investigação são os sistemas de PR com abordagens que buscam apoio em tecnologias semânticas para responder aos questionamentos dos usuários. Um sistema de PRS tem como principal objetivo gerar respostas a questionamentos formuladas em linguagem natural realizando consultas a BC semânticas como ontologias e Web Semântica (HÖFFNER et al., 2016).

2.3 Ontologia

O termo ontologia tem, pelo menos, duas acepções bastante diferentes dependendo da área de conhecimento em que é utilizado. Na Filosofia, por exemplo, é o ramo que estuda a existência das coisas, os tipos e estruturas dos objetos, suas propriedades, eventos e processos e suas relações com o mundo real (HOFWEBER, 2014).

Para a Inteligência Artificial (IA) o termo ontologia está associado a uma linguagem lógica formalmente definida com o objetivo de representar conhecimento. Na acepção da IA, a ontologia é então um artefato computacional (GUARINO; OBERLE; STAAB, 2009) que contém um conjunto de definições em um vocabulário formal para representar o conhecimento de um domínio de forma a ser possível o seu processamento por sistemas computacionais.

No que tange à área de pesquisa deste trabalho, a ontologia é um artefato computacional que especifica formalmente uma conceitualização consensualmente compartilhada por uma comunidade (BORST, 1997). Fundamentalmente, uma ontologia é

descrita por um conjunto de termos que representam entidades de um domínio e axiomas formais que impõem restrições a interpretação destes termos.

Em sendo um artefato computacional, a representação computacional da ontologia se dá através de sua representação em linguagens formalmente definidas, tais como a Ontolingua (GRUBER, 1992), a General Ontological Language (DEGEN et al., 2001), a Ontology Web Language (MCGUINNESS; VAN HARMELEN; OTHERS, 2004), entre outras.

A próxima seção apresenta uma visão geral da Ontology Web Language (OWL), uma vez que o sistema a ser desenvolvido nesta pesquisa tem como foco a integração com ontologias implementadas nesta linguagem.

2.3.1 OWL

Em 2000 a agência de defesa norte-americana DARPA financiou um projeto para o desenvolvimento da linguagem de representação de ontologias chamada DAML-ONT (HENDLER; MCGUINNESS, 2000). Em março de 2001 houve a fusão da DAML-ONT com a Ontology Inference Language (FENSEL et al., 2001), outra linguagem para escrita de ontologias, originando daí a linguagem DAML+OIL (HORROCKS; OTHERS, 2002). A linguagem OWL iniciou como uma pesquisa de revisão da DAML+OIL com vistas a ser o padrão de fato para a Web Semântica (BERNERS-LEE et al., 2001).

Vê-se então que a OWL é, na verdade, uma família de linguagens para representação do conhecimento utilizada para a criação de ontologias. A OWL despertou a atenção do mundo acadêmico e comercial quando a sua versão 2.0 foi oficialmente recomendada pela W3C em outubro de 2009 para ser utilizada na escrita de ontologias para a Web Semântica.

Na prática, uma ontologia escrita na linguagem OWL pode ser vista como um conjunto de axiomas que provêm uma lógica explícita de asserções especificamente sobre três elementos: Classes, Instâncias e Propriedades. Novos fatos, implicitamente descritos na ontologia, podem ser inferidos pelo uso de um sistema para raciocínio auxiliar chamado reasoner, o qual pode concluir novos relacionamentos entre os seus conceitos e instâncias.

Com relação a serialização em arquivos de uma ontologia OWL, foi definido pelo W3C que o padrão RDF/RDF-S (KLYNE; CARROLL, 2006) é a sintaxe concreta oficial e obrigatória para o OWL, devendo esta ser suportada por todos os sistemas que manipulam ontologias OWL.

2.3.2 RDF/RDF-S

A sintaxe da linguagem OWL é caracterizada pela semântica formal definida e pela serialização definida nos padrões RDF (Resource Description Framework) e RDF-S (RDF Schema) (DECKER et al., 2000). O RDF é um modelo de dados extremamente flexível básico utilizado para a representação de informações da Web Semântica. O modelo de dados do RDF não tem relação com o modelo de dados em tabelas dos bancos de dados relacionais e nem com as árvores de dados do XML, pois implementa um modelo baseado em grafos direcionados.

Figura 1: Exemplo esquemático de uma tripla RDF.

Fonte: elaborada pelo autor.

A Figura 1 apresenta a típica representação visual da tripla RDF: dois nós conectados um ao outro por uma aresta direcionada. Existem três diferentes tipos de nós:

1. Recursos: um nó do tipo recurso é qualquer coisa sobre a qual algo possa ser dito (por convenção, graficamente representada sempre por uma oval).
2. Literais: na prática, uma literal é sempre um valor (graficamente representada por um retângulo).
3. Blank nodes: um blank node é um recurso sem uma IRI (não representado na figura Figura 1).

Nós do tipo recurso e arestas são sempre representados por um Identificador de Recurso Internacional (IRI). Em relação a formação das triplas RDF, existem somente duas regras: nós do tipo recurso e blank nodes podem aparecer à direita ou a esquerda de uma aresta, enquanto literais somente à direita (MCBRIDE, 2004).

Isto significa que quaisquer recursos podem ser ligados através de uma aresta. Além desta flexibilidade de conexão entre os recursos, para a criação de novos recursos é suficiente criar-se uma nova IRI. Estes são os fundamentos básicos da Web Semântica, os quais lhe conferem uma flexibilidade ímpar em relação a outras tecnologias de representação de dados XML, bancos de dados relacionais, etc (DECKER et al., 2000).

2.3.3 Triple Store

Segundo definido pela W3C, a sintaxe concreta oficial para armazenamento de uma ontologia OWL é a RDF/XML (MCGUINNESS; VAN HARMELEN; OTHERS, 2004). Contudo, o armazenamento mais apropriado para a busca e recuperação eficiente de grandes quantidades de triplas é, sem dúvida, os bancos de dados especializados em triplas, chamados de triple stores (VOIGT; MITSCHICK; SCHULZ, 2012).

Existem diversas opções de plataformas triple stores disponíveis, como por exemplo AllegroGraph[2], Stardog[3], Virtuoso[4], GraphDB[5], entre outras. Estas plataformas apresentam diferentes níveis de desempenho, escalabilidade, API de programação e acesso. Alguns sistemas de triple store suportam inferência, transações e controle de acesso, mas com diferentes granularidades. Algumas plataformas têm seu foco em Big Data e escalam sobre clusters de computadores (SUN; JIN, 2010).

[2] https://franz.com/agraph/allegrograph
[3] https://www.stardog.com
[4] https://virtuoso.openlinksw.com
[5] https://www.ontotext.com/products/graphdb

A recuperação das triplas armazenadas nos sistemas de triple store pode ser realizada de variadas formas, utilizando desde comandos SQL padrões, pesquisas via indexação full-text, bem como suportando ao padrão de busca de triplas via comandos SPARQL.

2.3.4 SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) é a linguagem de consulta padrão oficialmente recomendada pelo W3C para a consulta de triplas RDF. O SPARQL (PRUD; SEABORNE, 2006) é tanto uma linguagem de consulta quanto um protocolo de comunicação.

Enquanto linguagem de consulta, o SPARQL está para as triple stores assim como o SQL está para os bancos de dados relacionais ou o XQuery está para o XML, com a diferença que o SPARQL foi projetado para operar ao mesmo tempo sobre fontes de dados locais e remotas.

É a função de protocolo do SPARQL permite a transmissão das consultas e resultados entre o cliente e o servidor, utilizando o HTTP como meio de comunicação. É esta característica do SPARQL que possibilita a busca em tempo real em endpoints SPARQL públicos. Um endpoint é somente um servidor que expõem seus dados via protocolo SPARQL.

O quadro 2 apresenta um exemplo de comando SPARQL para recuperar uma lista ordenada em ordem alfabética de nomes de uma base de dados do endpoint exposto em um servidor remoto contendo triplas como a apresentada na figura Figura 1.

Quadro 2: Exemplo de comando SPARQL.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
FROM <http://semantics.unisinos.br/dataset.rdf>
WHERE {
  ?x foaf:name ?name .
}
ORDER BY ?name
```

Junto aos modelos de metadados RDF, RDF-S e a linguagem OWL, o protocolo e linguagem de consultas SPARQL são as tecnologias fundamentais da Web Semântica.

3 TRABALHOS CORRELATOS

A questão de pesquisa proposta para esta tese definiu o contexto que norteou o levantamento do estado da arte, focando a busca em trabalhos recentes e relevantes, tendo todos em comum o fato de serem todos sistemas de PRS. Limitou-se a pesquisa a trabalhos publicados há no máximo 10 anos.

Os trabalhos apresentados neste capítulo são resultantes de pesquisas realizadas em sistemas especializados na busca de produções científicas, tais como o Scielo[6], Google Acadêmico[7], ACM Digital Library[8] e Scopus[9]. Utilizou-se as ferramentas de busca disponibilizadas nestes sistemas, utilizando como termos chaves de busca as seguintes palavras: short question answering, linked data, natural language interface e knowledge base.

Fez-se então uma avaliação preliminar do material retornado pelos sistemas de busca, tendo sido selecionados 168 artigos para leitura. A partir desta leitura, foram escolhidos oito diferentes abordagens, consideradas como de maior relevância para apresentação neste capítulo. Os trabalhos selecionados originam-se tanto de eventos quanto periódicos reconhecidamente importantes quanto ao seu valor científico para pesquisas na área da Computação.

Em relação a relevância, buscou-se selecionar para apresentação os trabalhos que influenciaram mais diretamente o desenvolvimento do trabalho aqui apresentado. Nos parágrafos seguintes serão explicitadas estas influências.

3.1 Wang, Ling e Hu

No trabalho apresentada por Wang, Ling e Hu (2019) propõe-se um modelo de PRS baseado em Rede Neural Artificial para localização de relações simples em BC. Este modelo é composto pelos módulos de associação de entidade e detecção de relações. Um vetor de embedding é calculado a partir da questão recebida para verificar a sua similaridade com entidades ou relações candidatas. A abordagem foca na representação da pergunta baseada em mecanismos de atenção para localização de relações simples por sistemas de PRS. No módulo de associação de entidade, dois métodos de agrupamento por mecanismo de atenção são utilizados, um focado nas características internas da sentença e outro na sua estrutura, ambos empregados para derivar o embedding das questões.

No módulo de detecção de relações propõe-se uma nova estrutura para o mecanismo de atenção chamada MLTA (Multilevel Target Attention), uma abordagem de agrupamento baseado em atenção que tem por objetivo utilizar as descrições multiníveis das relações. Nesta nova estrutura, os pesos do mecanismo de atenção para agregação dos estados ocultos das frases das questões são calculados usando as relações candidatas nas consultas aos embedding

[6] http://www.scielo.org
[7] https://scholar.google.com.br
[8] http://dl.acm.org
[9] https://www.scopus.com

em nível de carácteres, palavras e relações. A partir destas consultas, detecta-se a relação pelo cálculo dos escores de similaridade nos três níveis entre as questões e as relações candidatas.

A abordagem proposta tem por objetivo apresentar um modelo baseado somente em Rede Neural Artificial (fim a fim) para duas tarefas tipicamente realizadas por sistemas de PRS: identificação de entidades e relações em perguntas formuladas em LN. O foco na identificação de relações simples reduz significativamente a complexidade de análise do significado da frase.

3.2 WDAqua-core

O WDAqua-core (DIEFENBACH; SINGH; MARET, 2018) é um sistema baseado em regras que usa uma abordagem combinatorial para gerar consultas SPARQL a partir de questões em LN. Embora o sistema possa ser utilizado para processar sentenças em LN, ele ignora a sintaxe da questão, baseando-se exclusivamente na constituição léxica das palavras que compõem a pergunta, buscando a semântica para a interpretação da frase a partir do conteúdo da BC. Como o sistema restringe-se ao nível léxico das palavras, é agnóstico de linguagem.

A abordagem do sistema ENSEPRO é inspirada no sistema WDAqua-core, embora todos os processos tenham sido implementados com algoritmos próprios. Dentre as principais diferenças, destaca-se o fundamento em níveis mais elevados de abstração linguística para a interpretação da LN, utilizando os níveis morfológico, sintático e estrutural da frase para a predição do mapeamento com os elementos da BC.

3.3 Hamon, Grabar e Mougin

No trabalho desenvolvido por Hamon, Grabar e Mougin (2017) apresenta-se uma proposta para tradução de questões formuladas em LN em consulta SPARQL. São utilizadas ferramentas de Processamento de Linguagem Natural, recursos semânticos e descrições de tripas RDF. O método composto de quatro etapas que anota linguística e semanticamente as questões, para depois produzir uma abstração das questões a partir de regras que permitem relacionar os elementos obtidos com as entidades do contexto.

Após isto, o sistema conecta as entidades identificadas e constrói os padrões dos grafos a serem obtidos, os quais servirão de base para a formulação final da consulta. O método foi desenvolvido tendo por base 50 questões relativas às informações representadas em três bases de conhecimentos biomédicas utilizadas na tarefa 2 do desafio QALD-4 (UNGER et al., 2014) e avaliada sobre 27 novas questões. Analisando-se o F-measure de 0,78 obtido na avaliação, percebe-se que as abordagens baseadas em informações linguísticas das perguntas proporcionam uma boa infraestrutura para a implementação de sistemas de PRS.

3.4 SINA

Por ter sido desenvolvido inicialmente para realizar consultas a partir de palavras chaves, o sistema SINA (SHEKARPOUR et al., 2015) não estabelece as relações entre os

diferentes elementos da frase a partir da LN, mas sim a partir da BC. A questão é inicialmente segmentada, sendo então estabelecidas as associações entre as palavras da frase e os elementos da BC. Caso seja necessário, a desambiguação é realizada pelo modelo estatístico HMM[10] (STAMP, 2004).

Localizados os elementos da BC, o sistema SINA inicia a construção da consulta seguindo o seguinte fluxo de procedimentos: para cada instância ou classe, um vértice é criado. Para cada propriedade, uma aresta é criada. As arestas são usadas para conectar os vértices compatíveis entre si, ou seja, de acordo com o domínio e as restrições das propriedades. Caso não ocorra a compatibilidade, um ou dois vértices são criados, correspondendo a variáveis.

Este procedimento pode resultar em mais de um grafo, sendo todos eles considerados como consultas para localização da resposta, uma vez que não é possível determinar com clareza qual deles reflete a intenção do usuário. Mais que isto, ao fim do processo é possível que ocorram subgrafos desconectados. Neste caso, para cada par de vértices em dois subgrafos fixos busca-se o conjunto de propriedades possíveis de conectá-los. Todos os grafos possíveis são inseridos em uma consulta SPARQL a ser executada para localização das respostas.

Em relação a abordagem desenvolvida nesta tese, verifica-se que o sistema ENSEPRO também estabelece as relações entre as palavras a partir da BC, possibilitando assim o processamento de frases em LN ou somente palavras chaves. No entanto, diferentemente do sistema SINA, o sistema ENSEPRO não faz uso das restrições das propriedades para o estabelecimento das relações, possibilitando assim a sua aplicação sobre BC com pouca ou até mesma nenhuma restrição.

3.5 YodaQA

O YodaQA (BAUDIŠ, 2015) é um sistema de PRS híbrido de código aberto implementado sobre a plataforma Brmson[11], a qual é inspirada no DeepQA (o qual é visto na seção 3.9). O sistema YodaQA usa paralelização para processar as questões, o resultado da busca e os candidatos à resposta. O fluxo de processos do sistema está dividido em cinco etapas: (1) Análise da Questão, (2) Produção de Resposta, (3) Análise da Resposta, (4) Fusão e Ranqueamento de Respostas e, por fim, (5) Refino Sucessivo.

Na abordagem do YodaQA, assim como no DeepQA, a busca de respostas baseia-se na combinação de sites da internet, extração de informação, bancos de dados e Wikipédia em especial. A abordagem desenvolvida nesta tese também baseia-se num fluxo de processos bem definidos como a do YodaQA, divergindo em relação à base de conhecimento, uma vez que a abordagem aqui implementada visa buscar respostas unicamente em bases de conhecimento semânticas.

Assim como no sistema YodaQA, o sistema ENSEPRO também faz uso de processamento paralelo, mas especificamente na etapa de geração combinatorial e ranqueamento das respostas. A principal diferença entre as abordagens é o foco de aplicação.

[10] HMM é um acrônimo para *Hidden Markov Model* (Modelo Escondido de Markov).
[11] http://brmlab.cz/project/brmson/start

O sistema ENSEPRO foi projetado de forma a ser possível aplicá-lo em múltiplos domínios, inclusive concomitantes.

3.6 CASIA

O sistema CASIA (HE et al., 2014) usa Aprendizagem de Máquina em todo o processamento da pergunta e localização da resposta. Na etapa de análise a questão é segmentada para extrair suas características tais como a posição da frase, as etiquetas de POS, o tipo de dependência, entre outras. Na fase de mapeamento da frase, os elementos da BC são associados aos segmentos da sentença e novas características são extraídas, tais como o tipo do elemento e um escore de similaridade entre o segmento e o elemento da BC.

Na etapa de desambiguação, as características extraídas são usadas em uma rede lógica de Markov (MLN) para achar a relação mais provável entre os segmentos da frase, bem como os mais prováveis mapeamentos. As relações detectadas são então usadas para gerar a consulta SPARQL. A principal desvantagem da abordagem é a necessidade de retreinamento a cada nova BC.

Percebe-se atualmente uma tendência ao uso de técnicas baseadas em Aprendizagem de Máquina para a resolução de problemas em geral. Isto não é diferente no campo do PLN. Na abordagem desenvolvida nesta tese faz-se uso de embedding complementarmente ao uso das informações linguísticas. Aparentemente, as soluções que buscam fundamentar-se em uma técnica geralmente apresentam desempenhos medianos, como no caso do sistema CASIA (UNGER et al., 2014).

3.7 gAnswer

O sistema de PRS gAnswer (HUANG; ZOU, 2013) trata a ambiguidade das questões através de fragmentos RDF, partindo do princípio que a intenção de uma questão pode ser traduzida em um grafo. As variáveis, entidades e classes da questão são os nós do grafo, enquanto as relações são representadas por arestas. O grafo assim representado seria semanticamente equivalente à questão. As possíveis ambiguidades da questão seriam representadas por uma duplicidade de vértices e arestas do grafo. A ideia é desambiguar estes vértices e arestas procurando na base de conhecimento um subgrafo isomórfico cujos vértices correspondam probabilisticamente aos segmentos da questão. Por fim, os subgrafos que melhor casem com o grafo gerado são retornados.

A primeira avaliação da abordagem no QALD-3 mostrou que o mapeamento em grafos de questões em LN não era tão poderoso quanto gerar consultas SPARQL a partir de padrões preestabelecidos no que se refere a agregações e filtros, tendo o Xser obtido um fraco desempenho para responder várias questões desta edição do evento. Uma extensão do gAnswer (ZOU et al., 2014) foi lançada, obtendo as melhores marcas da 7ª e 9ª edições do evento QALD.

A abordagem do gAnswer é bastante diferenciada, tratando o problema de processamento da LN como uma questão de mapeamento léxico de palavras em elementos de grafos. Embora seja uma abordagem pouco intuitiva, os resultados obtidos nas avaliações dos

últimos três anos comprovam que a abordagem puramente léxica é uma solução que apresenta bons desempenhos no contexto de sistemas de PRS. A hipótese proposta nesta tese é que a eficiência destas abordagens pode ser melhorada pelo uso das informações linguísticas de nível de abstração mais elevadas.

3.8 Xser

O sistema Xser (XU; FENG; ZHAO, 2014) fundamenta-se na premissa de que os sistemas de PRS têm duas etapas básicas e independentes. Primeiramente, determinar a estrutura da questão, o que o Xser realiza com base em um grafo de níveis de dependência. Em um segundo momento, o sistema usa a base de conhecimentos para instanciar os padrões identificados. Com esta abordagem, pretende-se que a mudança da base de conhecimento afete somente parte do processo, diminuindo assim o esforço de conversão do sistema.

Na abordagem proposta no sistema Xser, associa-se uma lista de etiquetas semânticas às frases, reduzindo assim a questão a uma sequência de etiquetas que é submetida a uma rede neural artificial do tipo perceptron. Esta rede é treinada usando algumas características da questão, como n-gramas de etiquetas POS, anotações de sistemas de Reconhecimento de Entidades Nomeadas e palavras da frase.

O Xser usa o grafo de dependência da frase gerado por um parser treinado sobre um conjunto de dados manualmente anotado. O parser usa como características da questão as etiquetas de POS das palavras, o tipo da frase e a pergunta em si mesma. A vantagem é que o parser aprende automaticamente qual é a relação entre as frases. A grande desvantagem da abordagem recai no fato de que é necessário um corpus anotado manualmente.

3.9 DeepQA

Percebe-se nos últimos dez anos um maior número de diversos sistemas de PRS, como por exemplo o sistema DeepQA da IBM (GONDEK et al., 2012), o qual tornou-se famoso por jogar e vencer seres humanos no programa de auditório Jeopardy[12]. O sistema DeepQA trata perguntas complexas determinando primeiramente o elemento central da questão, o qual representa a entidade/resposta procurada. A informação sobre o elemento central é usada para predizer o tipo léxico da resposta, restringindo assim o conjunto de possíveis respostas. Esta abordagem permite o processamento de questões indiretas e múltiplas sentenças.

Uma forma diferente de selecionar o conjunto de candidatos e assim controlar a ambiguidade é determinar o tipo de resposta esperado. A abordagem padrão para determinar o tipo da resposta é identificar e mapear o foco da questão a uma classe da ontologia. Na questão “Quem escreveu o livro Os Pilares da Terra?”, por exemplo, o foco da questão é a palavra “livro”, a qual então é mapeada à classe dbo:Book[13].

Existe, no entanto, uma lista considerável de tipos de questões que não são facilmente mapeáveis. O sistema DeepQA utiliza um framework para coerção de tipos chamado TyCor (MURDOCK et al., 2012), gerando candidatos baseados em múltiplas interpretações e

[12] https://www.jeopardy.com
[13] O prefixo “dbo” é uma abreviatura para <http://dbpedia.org/ontology/Book>.

selecionando com base em uma combinação de escores. Além de tentar alinhar os tipos de resposta diretamente, pode também forçar o mapeamento pelo cálculo da probabilidade de uma entidade pertencente à classe A pertencer também à classe B.

Embora possa ser considerado como um marco importante em relação ao uso de sistemas de PRS, o código do DeepQA é fechado, impedindo a reprodução independente de experimentos de avaliação, ainda que os fundamentos do seu funcionamento tenham sido publicados em artigos. Analisando-se o material publicado, chega-se a conclusão de que a implementação do sistema DeepQA é para um domínio bastante específico, pois percebe-se que o seu foco foi a participação no programa Jeopardy, o que compromete o seu uso em outras aplicações.

3.10 FREyA

O sistema FREyA (DAMLJANOVIC; AGATONOVIC; CUNNINGHAM, 2010) é um trabalho fortemente relacionado ao alvo de pesquisa desta proposta, por ser definido como uma abordagem de interface em linguagem natural para BC semânticas. Baseia-se em técnicas de PLN, tais como a análise sintática e listas de gazetteer, para realizar a busca de respostas em ontologias.

O sistema apoia-se no vocabulário de ontologias como fonte primária para a identificação de conceitos em questões expressas em linguagem natural, utilizando a análise sintática obtida com o uso de parser linguísticos para ligar elementos da frase (POC – Potential Ontology Concepts) a conceitos da ontologia (OC – ontology concepts).

As palavras usadas na questão (POC) são associadas aos conceitos da ontologia (OC) em um processo chamado de consolidação. Caso identifique-se uma lista de OC para uma POC (ambiguidade), dispara-se um processo de interação com o usuário visando a desambiguação. Caso reste alguma POC sem associação, aciona-se o usuário para realizar o mapeamento manualmente.

O sistema FREyA faz uso de algumas ferramentas da suíte de processamento de linguagem natural GATE (CUNNINGHAM et al., 2002), como por exemplo o parser linguístico para geração de informações sintáticas e árvore de dependências, bem como o OntoRoot Gazeteer[14] para geração dinâmica de listas de nomes de locais geográficos.

O mapeamento entre os elementos presentes na questão elaborada pelo usuário e os conceitos presentes na ontologia onde será realizada a busca da resposta é derivado das informações sintáticas e da árvore de dependências geradas por um parser linguístico, combinadas com a lista de nomes de locais geográficos gerada pelo OntoRoot, complementados por um sistema de ranqueamento baseado em um conjunto de regras expandido pelo uso da ontologia Cyc[15] e dos synsets da Wordnet[16]. Possíveis erros de ortografia do usuário são minimizados pelo uso combinado das métricas de Monge Elkan (COHEN; RAVIKUMAR; FIENBERG, 2003) com o algoritmo Soundex (HOLMES; MCCABE, 2002).

[14] https://gate.ac.uk/sale/tao/splitch13.html
[15] http://www.cyc.com/platform/enterprisecyc-license
[16] https://wordnet.princeton.edu

Uma vez mapeados os termos da questão do usuário nos conceitos da ontologia, é gerado um comando SPARQL para realizar a busca dos dados. A fim de definir como apresentar ao usuário a resposta encontrada na consulta é realizada uma classificação do tipo de questionamento realizado pelo usuário através da aplicação de uma série de regras.

O desempenho do sistema foi avaliado utilizando-se os dados do Geoquery Data[17]. A implementação do sistema compreende 3 fases principais: (1) identificação e verificação dos conceitos da ontologia, (2) geração da consulta SPARQL e (3) identificação do tipo de resposta a ser apresentada ao usuário.

Pode-se observar que a abordagem proposta nesta tese tem pontos em comum com o sistema FREyA, como a fundamentação em informações linguísticas da pergunta para busca de elementos da ontologia. Contudo, pode-se perceber que a seleção do POC do FREyA fundamenta-se (tanto quanto descrito) na classe das palavras, desconsiderando as estruturas linguísticas, o que implica em uma maior necessidade do acionamento do usuário. Aliás, o acionamento do usuário é outro diferencial, uma vez que a proposta aqui apresentada conta somente com a eficiência do sistema de ranqueamento para realizar a devida desambiguação.

3.11 Considerações finais sobre os trabalhos correlatos

A análise realizada neste trabalho sobre os sistemas de PRS mostra que muitos sistemas fazem uso de técnicas semelhantes no decorrer de sua execução. Verifica-se, por exemplo, que muitos sistemas usam abordagens semelhantes na análise da questão e na tarefa de mapeamento da frase. No entanto, vê-se também evidenciado que os sistemas de PRS em geral concentram-se no uso de algumas técnicas em detrimento de outras.

Alguns trabalhos evidentemente focam sua atenção em técnicas de Aprendizagem de Máquina, deixando de lado as informações linguísticas, enquanto outros fazem uso de informações linguísticas, ignorando os recursos baseados em Aprendizagem de Máquina.

Verifica-se também que as abordagens que fazem uso de informações linguísticas (BAUDIŠ, 2015; DAMLJANOVIC; AGATONOVIC; CUNNINGHAM, 2011; GONDEK et al., 2012; HAMON; GRABAR; MOUGIN, 2017) geralmente restringem-se ao uso das etiquetas POS, explorando de forma menos aprofundada as estruturas linguísticas que carregam um alto valor semântico para a interpretação da pergunta. Pode-se também observar que as implementações que optam por técnicas baseadas em PLN estatístico (HE et al., 2014; SHEKARPOUR et al., 2015; WANG; LING; HU, 2019) ou em grafos (XU; FENG; ZHAO, 2014; ZOU et al., 2014) geralmente ignoram as demais informações linguísticas da frase, deixando de lado uma fonte de informações significativa para interpretação da semântica da pergunta.

Ainda em relação á técnica de desenvolvimento, pode-se observar em surveys recentes um número considerável de pesquisas em andamento buscando a aplicação de Aprendizagem de Máquina como base para o desenvolvimento de sistemas de PRS (DIEFENBACH et al., 2018b; HÖFFNER et al., 2017). Tais trabalhos, sejam elas baseadas na ultra recente Aprendizagem Profunda (WANG; LING; HU, 2019) ou em técnicas mais tradicionais, como aquelas baseadas nos Modelos de Markov (SHEKARPOUR et al., 2015). Em relação aos

[17] http://www.cs.utexas.edu/users/ml/nldata.html

objetivos do trabalho aqui apresentado, a principal desvantagem das abordagens baseadas em AM recai sobre a necessidade de corpus anotado para treinamento do modelo.

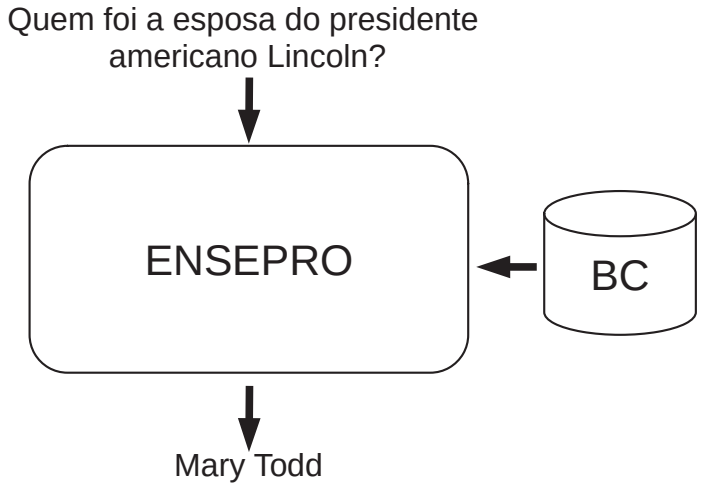
Além disto, os modelos baseados em AM ainda não apresentam desempenho que se destaquem nos eventos mais recentes de avaliação específicos da área (NGOMO, 2019; USBECK et al., 2018), principalmente se comparados às abordagens baseadas em grafos (DIEFENBACH; SINGH; MARET, 2018; ZOU et al., 2014). Se por um lado os sistema de PRS baseados em grafo apresentam-se como o estado da arte, observa-se uma tendência a apresentar sistemas independentes de idioma, restringindo-os aos níveis léxico e morfológico, o que limita a sua capacidade de compreensão da LN.

Esta tese tem por objetivo apresentar um modelo que combine de forma inovadora as vantagens das três principais técnicas de implementação de sistemas de PRS citadas acima. Neste sentido, propõe-se primeiramente um aprofundamento na utilização das informações linguísticas nas várias etapas do processo de análise da pergunta, explorando de forma inovadora as estruturas linguísticas para inferência do significado da LN. Esta abordagem inicial, combinada com a busca em grafo para a geração e ranqueamento de candidatas e, por fim, apoiada pelo uso de embedding para seleção e validação das respostas, resulta em um sistema de PRS baseado em regras com generalização suficiente para responder perguntas curtas independente de domínio formuladas em português. O próximo capítulo tem por objetivo apresentar os detalhes deste modelo.

4 ENGENHO SEMÂNTICO DE PERGUNTA E RESPOSTA ORIENTADO A ONTOLOGIAS

Esta seção tem como objetivo apresentar o modelo do sistema de PRS chamado ENSEPRO, abreviatura para Engenho Semântico de Perguntas e Respostas para Ontologias. O sistema ENSEPRO tem como objetivo receber perguntas formuladas em LN e realizar a busca da respectiva resposta em BC semântica. A Figura 2 apresenta uma visão geral deste modelo.

Figura 2: Visão geral do funcionamento do sistema ENSEPRO.



Fonte: elaborada pelo autor.

A abordagem do modelo fundamenta-se na premissa de que informações linguísticas presentes nas questões em LN combinadas com o uso de busca em grafos e embedding possibilitam a implementação de um sistema de PRS independente de domínio cujo desempenho mostra-se adequado em relação ao estado da arte atual.

Considera-se inicialmente as informações morfológicas e sintáticas das palavras que compõem a pergunta em LN, bem como as informações estruturais da frase em si. Com base nestas informações linguísticas é possível otimizar os processos necessários para a compreensão da pergunta, possibilitando, por exemplo, predizer os possíveis papéis que as palavras da frase em LN podem assumir nas triplas que compõem a BC semântica. Esta e outras utilizações das informações linguísticas são apresentadas e discutidas no decorrer deste capítulo.

A partir da melhor compreensão dos termos que compõem a pergunta em linguagem natural, o sistema ENSEPRO faz uso da técnica de busca e geração de triplas candidatas a respostas utilizando uma abordagem baseada em busca em grafo.

A principal vantagem do uso das informações linguísticas combinada à técnica de grafo é a diminuição do esforço computacional para seleção e geração das respostas

encontradas pela busca em grafo, utilizando as informações sintáticas dos termos relevantes para a predição dos papeis que o termo relevante pode assumir nas triplas.

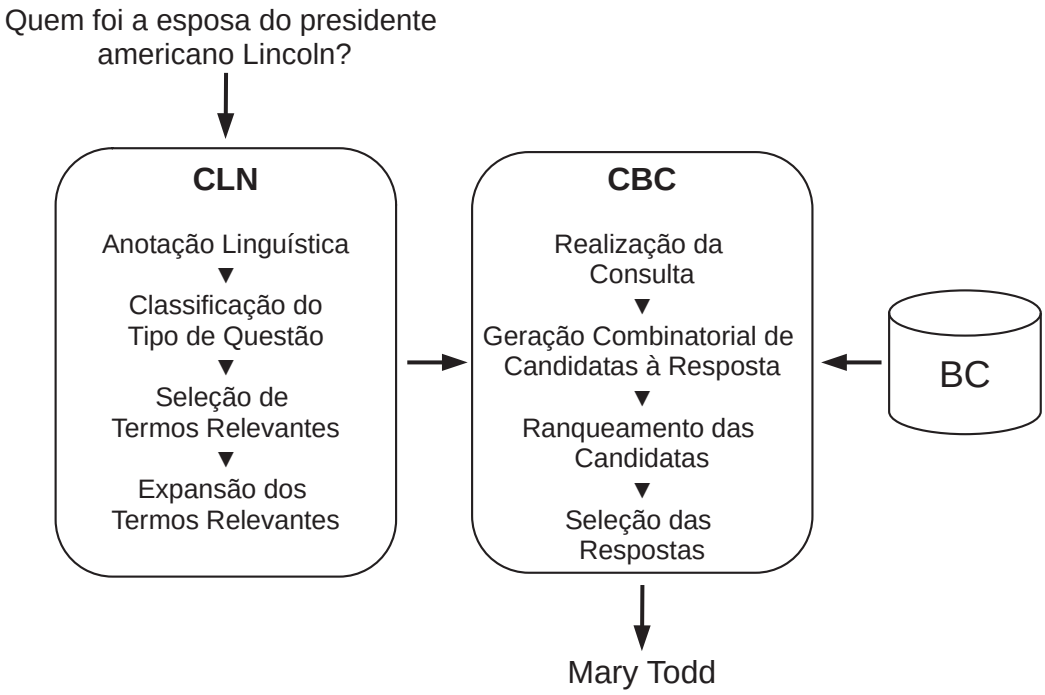
O embedding é utilizado como um apoio no momento de selecionar as respostas corretas. Caso a abordagem baseada em ranqueamento linguístico não seja suficiente para discernir se as respostas encontradas relacionam-se à pergunta, utiliza-se o word embedding para verificar a similaridade entre a resposta encontrada (tripla) e as palavras chaves da frase (termos relevantes). A semântica latente do word embedding possibilita que o algoritmo de seleção não mais dependa da similaridade léxica entre os termos relevantes e os elementos que compõem as triplas.

O uso combinado das técnicas de busca em grafo, informações linguísticas e word embedding do modelo são descritos em detalhes nas seções 4.3 A Consulta à Base de Conhecimento e 4.3.5 Seleção de respostas, Contudo, antes de adentrarmos aos detalhes, é importante apresentar-se uma visão mais geral do modelo proposto.

4.1 Visão Geral do Modelo

O modelo proposto compõe-se de duas etapas principais: a Compreensão da questão em Linguagem Natural (CLN) e a Consulta à Base de Conhecimentos (CBC) para identificação das respostas. Cada uma destas etapas é composta respectivamente por quatro subprocessos sequenciais, os quais são apresentados na Figura 3.

Figura 3: Visão geral do fluxo de processos do sistema ENSEPRO.



Fonte: elaborada pelo autor.

Resumidamente, a primeira etapa do processo busca a Compreensão da pergunta em Linguagem Natural (CLN), realizada inicialmente pela anotação linguística da frase. As anotações linguísticas são as informações fundamentais para o funcionamento do sistema, sendo utilizadas nos demais processos do CLN, como na classificação do tipo da questão e na seleção dos termos relevantes (TR) da pergunta[18].

Após a anotação linguística, realiza-se a classificação do tipo de pergunta. O tipo de pergunta é utilizado pelo módulo CBC na interpretação dos termos relevantes da frase. Além disso, o tipo de pergunta será utilizado também no momento de gerar a resposta em linguagem natural ao usuário, processo atualmente fora do escopo do trabalho aqui apresentado

Classificado o tipo da pergunta, inicia-se a busca dos termos relevantes da pergunta. O processo de seleção dos termos relevantes tem por objetivo localizar as palavras da pergunta que são importantes para a localização da resposta. É a partir dos termos relevantes que define-se as consultas que serão realizadas na BC.

São também utilizados na consulta um conjunto de palavras derivadas pelo processo de expansão dos termos relevantes. A geração deste conjunto de palavras derivadas tem por objetivo compensar uma possível diferença de entre as palavras escolhidas pelo usuário no momento de formular a pergunta e o vocabulário definido para a representação das informações na BC. Este é o último processo realizado pelo módulo CLN.

Os termos relevantes e seus derivados são utilizados no primeiro processo do módulo de Consulta à Base de Conhecimento (CBC) para a busca de todas as possíveis triplas candidatas à resposta da pergunta.

O resultado deste processo de consulta à BC é uma lista de triplas, as quais são agrupadas no processo de geração combinatorial de triplas candidatas à resposta. O objetivo deste processo é gerar as possíveis combinações de triplas candidatas para responder perguntas mais complexas.

Uma vez finalizado o processo de geração de candidatas, as triplas são submetidas a um processo de ranqueamento, cujo o objetivo é possibilitar ao módulo CBC a identificação das respostas à questão recebida pelo sistema.

O último processo realizado pelo módulo CBC é a seleção das respostas entre as triplas candidatas ranqueadas. É neste processo que, dependendo do resultado obtido pelo ranqueamento, utiliza-se word embedding para seleção ou validação das respostas.

4.2 A Compreensão da Linguagem Natural

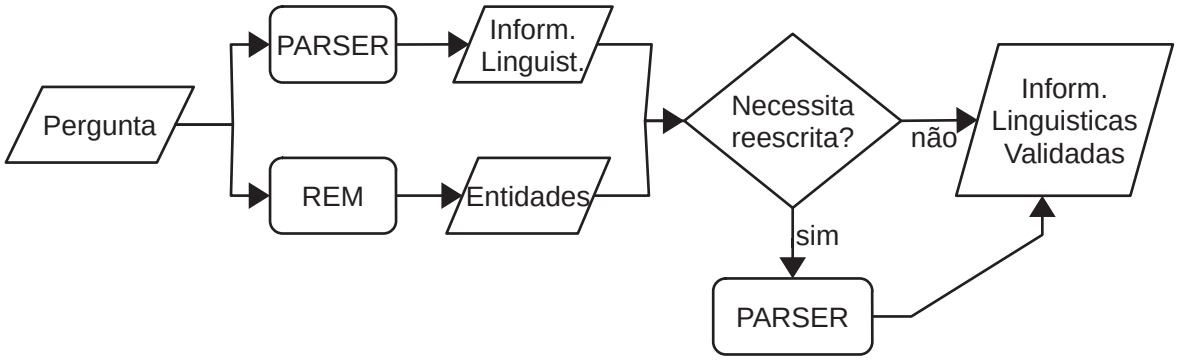
Para que um sistema de PRS encontre a resposta à questão é necessário primeiramente haver um processamento prévio da pergunta, de forma que permita ao sistema obter algum nível de compreensão da pergunta feita. Na abordagem proposta neste trabalho, parte-se da hipótese de que as informações linguísticas da pergunta auxiliam no processo de compreensão do que o usuário perguntou e o que se espera como resposta. Este é o motivo pelo qual o primeiro processo do CLN é a anotação linguística da frase.

[18] Chama-se de termo relevante àquelas palavras da pergunta que são essenciais para a localização da resposta na Base de Conhecimento.

4.2.1 Anotação linguística da pergunta

A etapa de anotação linguística da pergunta constitui-se da submissão da pergunta primeiramente a um analisador linguístico profundo[19] (parser linguístico) e posteriormente a um sistema de reconhecimento de entidades mencionadas (REM). O objetivo do envio da pergunta aos parser e ao sistema REM é obter-se um conjunto de informações linguísticas mais precisas, o que é alcançado pela validação dos substantivos próprios. A Figura 4 apresenta uma visão esquemática geral do algoritmo de validação das informações linguísticas.

Figura 4: Processo de validação das anotações linguísticas.



Fonte: elaborada pelo autor.

Inicialmente a pergunta é submetida ao parser linguístico, o qual analisa a frase e retorna as informações morfossintáticas das palavras que compõem a pergunta, bem como algumas informações estruturais da frase.

O parser retorna uma ampla gama de informações linguísticas, as quais são utilizadas no decorrer do processo da etapa de CLN. Nesta fase do processo tem-se especial atenção aos substantivos próprios identificados pelo parser na pergunta. Os substantivos próprios são, se presentes na frase, elementos chaves para o sistema ENSEPRO, tanto para a compreensão da pergunta quanto para a localização da respectiva resposta.

A importância dos substantivos próprios advém principalmente do significado que estas palavras carregam, ou seja, o seu valor semântico. Na linguagem natural, os substantivos próprios são as palavras utilizadas para representar os seres, entidades e conceitos do mundo real, como por exemplo Abraham Lincoln (pessoa), Mary Todd (pessoa), Estados Unidos da América (entidade), etc.

No contexto do sistema ENSEPRO, os substantivos próprios sempre denotarão um marco referencial de grande importância para a compreensão da pergunta e localização da resposta, devido principalmente ao seu papel semântico já bastante claro na frase. Além disto, no contexto do sistema ENSEPRO, os substantivos próprios possibilitam reduzir significativamente a quantidade de informações a serem processadas durante o processo, sendo fundamentais no momento de selecionar as respostas na etapa de consulta à BC.

[19] Para mais informações sobre quais são as características de um parser linguístico profundo, veja {*FONTE DE INFORMAÇÃO*}.

No Quadro 3 apresenta-se, a título de exemplo, as informações linguísticas retornadas pelo parser para a frase “Quem foi a esposa do presidente americano Lincoln?”. Neste exemplo há somente um substantivo próprio, o qual é identificável pela etiqueta “prop” (linha 16) atribuída pelo parser à palavra “Lincoln”.

```
1. SOURCE: Running text
2.  1. Quem foi a esposa do presidente americano Lincoln?
3.  A1
4.  QUE:fcl
5.  =Cs:spec("quem" <clb> <*> <interr> M/F S/P)  Quem
6.  =P:v-fin("ser" <fmc> <mv> PS 3S IND VFIN)   foi
7.  =S:np
8.  ==DN:art("o" <artd> F S)a
9.  ==H:n("esposa" <Hfam> F S)   esposa
10.  ==DN:pp
11.  ===H:prp("de" <sam-> <np-close>)        de
12.  ===DP:np
13.  ====DN:art("o" <-sam> <artd> M S)      o
14.  ====H:n("presidente" <Hprof> M S)      presidente
15.  ====DN:adj("americano" <nat> <jh> <np-close> M S)     americano
16.  ====DN:prop("Lincoln" <hum> <*> <np-close> M S) Lincoln
17. =?
```

Quadro 3: Informações linguísticas geradas pelo parser ao analisar a frase "Quem foi a esposa *do presidente americano Lincoln?*”, tendo destacado em negrito o substantivo próprio.

O algoritmo de identificação de substantivos próprios do parser linguístico faz uso de uma heurística bastante simplificada: se houver na frase uma palavra (ou sequência de palavras, no caso de nomes próprios compostos) que esteja escrita com ao menos a primeira letra maiúscula, desde que não esteja no início da frase e não seja prévia e usualmente reconhecida como pertencente a uma classe gramatical específica, então esta palavra (ou sequência) será identificada como um substantivo próprio.

Esta é uma heurística válida, mas a possibilidade de falha é bastante grande, uma vez que o algoritmo considera somente a grafia da palavra para classificá-la. Em nossa abordagem realiza-se uma verificação adicional em relação aos substantivos próprios, validando-se esta classificação pela submissão da frase a um sistema de REM.

4.2.2 Submissão da pergunta ao sistema de REM

Dada a importância dos substantivos próprios para o funcionamento do ENSEPRO, bem como a simplicidade do algoritmo de identificação implementado no parser linguístico, faz-se necessário verificar e validar se a classificação do parser está correta. A validação dos substantivos próprios é realizada pela submissão da pergunta a um sistema especializado no Reconhecimento de Entidades Mencionadas (REM).

O sistema REM buscam identificar nas frases as palavras que referem-se a entidades do mundo real. Como dito anteriormente, os substantivos próprios referenciam nomes de indivíduos/conceitos do mundo real. Sendo assim, faz-se uso da capacidade de identificar

entidades dos sistemas REM para verificar e confirmar os substantivos próprios identificados pelo parser.

Caso o sistema REM identifique uma entidade na pergunta, valida-se se há coerência entre o que foi identificado pelo parser como sendo um substantivo próprio e as referências a entidades identificadas pelo REM. Caso não haja concordância entre o parser e o sistema de REM, podem-se realizar algumas adequações na escrita da frase para que o parser produza as anotações linguísticas conforme o retorno do sistema de REM.

Caso a frase tenha sofrido alguma alteração devido a conflitos entre o parser linguístico e o sistema REM, ela é submetida novamente ao parser para análise. Esta nova submissão ao parser linguístico é necessária pois a frase pode sofrer alterações bastante profundas, o que pode modificar substancialmente a interpretação da frase e, por consequência, influenciar a compreensão da pergunta.

Para uma melhor compreensão do procedimento de validação dos substantivos próprios pela verificação da coerência entre o parser linguístico e o sistema REM,

Segue um exemplo prático do processo de validação dos substantivos próprios pela verificação da coerência entre o parser linguístico e o sistema REM para a frase “Quem foi a esposa do presidente americano Lincoln?”. Conforme já visto anteriormente, o parser linguístico classifica a palavra Lincoln como um substantivo próprio. Para a validação do substantivo próprio, a frase é então submetida ao sistema REM, obtendo-se o retorno apresentado no Quadro 4.

```
1. {
2.    "@text": "Quem foi a esposa do presidente americano Lincoln?",
3.    "@confidence": "0.7",
4.    "@support": "0",
5.    "@types": "",
6.    "@sparql": "",
7.    "@policy": "whitelist",
8.    "Resources": [
9.       {
10.          "@URI": "http://pt.DBPedia.org/resource/Abraham_Lincoln",
11.          "@support": "715",
12.          "@surfaceForm": "Lincoln",
13.          "@offset": "42",
14.          "@similarityScore": "0.999088384766567",
15.       }
16.    ]
17. }
```

Quadro 4: Informações retornadas pelo sistema REM ao analisar a frase "Quem foi a esposa *do presidente americano Lincoln?*”.

O sistema REM identificou que a palavra Lincoln é uma referência a uma entidade do mundo real, o que então confirma a classificação feita pelo parser. No entanto, ocorre para este exemplo uma situação bastante comum no processo de validação: a palavra Lincoln é associada pelo sistema REM a uma URI específica (linha 10 do Quadro 4).

A URI retornada pelo sistema REM indica que a representação do substantivo próprio Lincoln é feita na BC através do termo Abraham_Lincoln. A partir desta informação retornada pelo REM, o sistema ENSEPRO substitui o TR original da frase (Lincoln) pela entidade reconhecida (Abraham_Lincoln).

No Quadro 5 pode-se verificar este processo de substituição do TR ao executar-se o sistema ENSEPRO com as opções para listar os TRs originais (linha 1), selecionados a partir do retorno dado pelo parser (linha 6 a 9). Executando-se o sistema ENSEPRO com a opção de visualização da listagem de TRs na sua forma final (linha 10), pode-se verificar que a chamada ao sistema REM tem como efeito a substituição do TR “Lincoln” (linha 9) por “Abraham_Lincoln” (linha 17).

1. # ensepro **-tr -original** -frase "Quem foi a esposa do presidente americano Lincoln?"
2. Analisando frase(s)...
3. -> Frase 1: Quem foi a esposa do presidente americano Lincoln?
4. --> Tipo: TipoFrase={tipo=quem, confianca=0.89, ids=[2]}
5. --> Voz: Voz.ATIVA
6. --> Termos Relevantes:
7. ----> TR 0: [H:n|6| esposa]
8. ----> TR 1: [H:n|11| presidente]
9. **----> TR 2: [DN:prop|13| Lincoln**]

10. # ensepro **-tr -final** -frase "Quem foi a esposa do presidente americano Lincoln?"
11. Analisando frase(s)...
12. -> Frase 3: Quem foi a esposa de o <split> Abraham_lincoln <split>
13. --> Tipo: TipoFrase={tipo=quem, confianca=0.69, ids=[2]}
14. --> Voz: Voz.ATIVA
15. --> Termos Relevantes:
16. ----> TR 0: [H:n|6| esposa]
17. **----> TR 1: [H:prop|11| Abraham_lincoln**]

Quadro 5: retorno da execução do sistema ENSEPRO para visualização da influência do sistema REM na identificação dos substantivos próprios da pergunta.

Para demonstrar a importância deste mecanismo de validação dos substantivos próprios para o funcionamento do ENSEPRO, apresenta-se no Quadro 6 as respostas retornadas pelo sistema com (linha 1) e sem (linha 3) o uso do sistema REM. Percebe-se que o sistema REM foi essencial para a para a localização da resposta correta (linha 2). As respostas listadas para a execução sem o sistema REM (linhas 4 a 8) exigiriam do sistema ENSEPRO um esforço adicional para identificar que a resposta esperada é somente “mary todd”.

1. # ensepro **-resposta** -frase "Quem foi a esposa do presidente americano **Lincoln**?"
2. {'subject': '**abraham_lincoln**', 'predicate': 'conjuge', 'object': 'mary todd'}

3. # ensepro **-resposta** -frase "Quem foi a esposa do presidente americano **Lincoln**?"
4. {'subject': 'abraham_**lincoln**', 'predicate': 'conjuge', 'object': 'mary todd'}
5. {'subject': 'andrew_**lincoln**', 'predicate': 'conjuge', 'object': 'gael anderson'}
6. {'subject': 'e.k._**lincoln**', 'predicate': 'conjuge', 'object': 'ada olive proctor'}
7. {'subject': 'elmo_**lincoln**', 'predicate': 'conjuge', 'object': 'ida lee tanchick'}
8. {'subject': 'elmo_**lincoln**', 'predicate': 'conjuge', 'object': 'sadie whited'}

Quadro 6: respostas retornadas pelo sistema ENSEPRO quando executado com e sem o sistema REM ativado.

Importante salientar que, embora tenha sido implementado no ENSEPRO um algoritmo que vai identificar e tratar situações de múltiplas candidatas a resposta como a apresentada nas linhas 3 a 8 do Quadro 6 (detalhes sobre o funcionamento deste algoritmo serão vistos seção 4.3.5 Seleção de respostas), pode-se verificar neste exemplo que o uso do sistema REM teve como efeito a diminuição do esforço computacional para a identificação da resposta correta.

Além do menor esforço computacional, a submissão da frase ao sistema REM pode ser decisivo para a correta interpretação da pergunta em alguns casos. No exemplo o apresentado no Quadro 7 pode-se observar que, devido a uma falha no algoritmo do parser linguístico, o trecho “O Grito de Munch” é considerado como um único substantivo próprio (linha 5). Este erro de interpretação do parser faz com que o sistema ENSEPRO não identifique corretamente os TRs da pergunta, o que resulta na impossibilidade de localização da resposta correta para a questão.

1. # ensepro -frase "Em que museu está exposto O Grito de Munch?" -tr -original
2. --> Termos Relevantes:
3. ----> TR 0: [H:n|6| museu]
4. ----> TR 1: [H:v-pcp|9| exposto]
5. ----> TR 2: [DN:prop|10| O_Grito_de_Munch]

6. # ensepro -frase "Em que museu está exposto O Grito de Munch?" -tr -final
7. Frase: Em que museu está exposto O_grito <split> de <split> Edvard_munch?
8. --> Termos Relevantes:
9. ----> TR 0: [H:n|6| museu]
10. ----> TR 1: [H:v-pcp|9| exposto]
11. ----> TR 2: [H:prop|11| O_grito]
12. ----> TR 3: [DP:prop|14| Edvard_munch]

Quadro 7: exemplo de frase em que o sistema REM é utilizado para corrigir um erro de interpretação do parser linguístico quanto à identificação dos substantivos próprios.

Ao submeter-se a frase ao sistema REM, este identifica que “O Grito” e “Munch” são referências a duas entidades diferentes. A partir deste retorno do REM, o sistema ENSEPRO

atua sobre a frase, inserindo marcas (linha 7 do Quadro 7) que vão ajudar o parser a identificar corretamente os dois substantivos próprios (linhas 11 e 12).

Pode-se verificar então que o sistema REM cumpre com dois papéis fundamentais em relação aos TRs que são substantivos próprios: prioritariamente confirmando os substantivos próprios encontrados pelo parser, mas também definindo a forma a ser utilizada na busca da informação na BC. O acionamento do sistema REM para validar os substantivos próprios apontados pelo parser possibilita ao sistema ENSEPRO realizar buscas mais precisas, aumentando substancialmente a probabilidade de encontrar a resposta correta para a pergunta feita.

A validação dos substantivos próprios é o último processo da etapa de anotação linguística da questão. As informações linguísticas produzidas nesta etapa do CLN são utilizadas no decorrer de todo o fluxo do sistema ENSEPRO, inclusive no processo de classificação do tipo da questão.

4.2.3 Classificação do tipo de questão

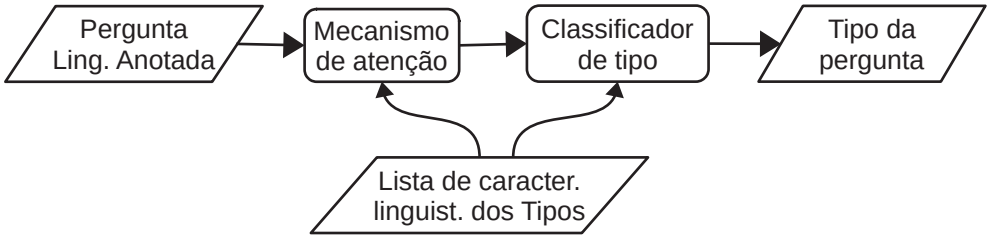
A análise da questão para classificação do tipo de pergunta é um processo comumente realizado em sistemas de Pergunta e Resposta (HERMJAKOB, 2001). A identificação do tipo de pergunta recebida pelo sistema pode ser usado para, por exemplo, determinar que tipo de resposta o usuário espera receber como retorno.

Além disto, o tipo de pergunta pode ser utilizado para, no momento de realizar a busca da resposta na BC, determinar alguns componentes adicionais da consulta. A influência do tipo de questão na consulta à BC será vista em maiores detalhes na seção 4.3.1 Consulta à Base de Conhecimento. Nesta seção serão vistos somente os detalhes da abordagem desenvolvida para a classificação da pergunta.

Existem diversas abordagens possíveis para realizar a classificação do tipo de pergunta (SILVA et al., 2011). Avaliando-se a complexidade, o custo computacional frente ao desempenho relatado das opções disponíveis (METZLER; CROFT, 2005), decidiu-se pelo desenvolvimento de uma abordagem própria (DE ARAUJO; HENTGES; RIGO, 2018), a qual apresentou um desempenho satisfatório em experimentos de avaliação realizados.

O algoritmo de classificação da sentença recebida pelo sistema ENSEPRO utiliza as anotações linguísticas geradas pelo parser para determinar o tipo da pergunta. O fluxo do processo de classificação da pergunta está esquematicamente representado na Figura 5.

Figura 5: Fluxo de processos do algoritmo de classificação do tipo de pergunta baseado em informações linguísticas.



Fonte: elaborada pelo autor.

O processo constitui-se da submissão da pergunta linguisticamente anotada a um algoritmo que realiza a remoção de todos os elementos da frase que não são relevantes para a classificação do seu tipo. Este processo de remoção tem por objetivo deixar na frase somente os elementos que são efetivamente utilizados para a classificação da pergunta.

Uma vez que as palavras irrelevantes para a classificação (e as suas respectivas anotações linguísticas) são removidas da frase, deu-se a este processo o nome de “Mecanismo de Atenção”, pois após a sua execução permanecem na frase somente os elementos essenciais para a classificação do tipo, ou seja, os elementos que “merecem atenção” do classificador.

A importância da palavra (ou seja, se ela será ou não removida da frase) é avaliada com base na Lista de Características Linguísticas dos Tipos de Perguntas (Quadro 8). A lista compõe-se de um dicionário (linhas 1 a 15) e um mapa (linhas 16 a 30) de palavras e anotações linguísticas. No dicionário estão as palavras importantes para a classificação e as respectivas anotações linguísticas, enquanto no mapa estão os tipos de perguntas e as respectivas sequências de palavras.

Quadro 8: Lista das Características Linguísticas contendo o dicionário e o mapa dos Tipos de Perguntas utilizado pelo Mecanismo de Atenção e pelo algoritmo de classificação.

```
1.  "dicionario": {
2.     "quem": "#quem <clb> <interr> M/F",
3.        "quanto": "#quanto <interr>  DET",
4.        "quando": "#quando <clb> <interr>",
5.        "que": "#que <clb> <interr> DET",
6.        "onde": "#onde <clb> <interr>",
7.        "aonde": "#aonde <clb> <interr>",
8.        "listar": "#listar <vH> <mv>",
9.        "buscar": "#buscar <vH> <mv>",
10.       "achar": "#achar <vH> <mv>",
11.       "encontrar": "#encontrar <mv>",
12.       "em": "#em <clb> <*>",
13.       "dia": "#dia <temp> <dur> M S",
14.       "vez": "#vez <temp> F P",
15.       }
16. "mapeamento": {
17.   "quem": [
18.     "$quem" ],
19.   "em_que": [
20.     "$em $que" ],
21.   "quando": [
22.     "$quando",
23.     "$em $que $dia"],
24.   "onde": [
25.     "$onde",
26.     "$aonde" ],
27.   "consulta": [
28.     "$listar",
29.     "$buscar" ]
30. }
```

O Mecanismo de Atenção verifica no dicionário (linha 1 do Quadro 8) quais palavras permanecem ou são removidas da pergunta. Caso a palavra não se encontre no dicionário, ela

é descartada por ser considerada como irrelevante para a classificação do tipo da pergunta. A partir da remoção das palavras irrelevantes simplifica-se consideravelmente o trabalho/esforço computacional necessário para o classificador definir o tipo da questão.

Removidas as palavras irrelevantes, a frase é submetida ao classificador, o qual vai utilizar o mapa (linha 16 do Quadro 8) para classificar o tipo da pergunta. É interessante observar que a estrutura de dados do mapa foi projetada de forma a ser flexível o bastante para representar todos tipos de perguntas atualmente identificados (linhas 17, 19, 21, 24 e 27).

Os tipos de perguntas podem compor-se de uma palavra somente (linha 18), ou uma sequência de palavras (linha 20), ou uma lista de palavras (linhas 25, 26, 28 e 29) ou ainda uma lista de palavra e sequências combinadas (linhas 22 e 23).

A classificação do tipo da pergunta é fundamentado na identificação da sequência de anotações linguísticas associadas ao tipo na frase da pergunta mais semelhantes às anotações linguísticas das palavras remanescentes após o processo de seleção do Mecanismo de Atenção.

A verificação da semelhança é realizada utilizando-se a distância de Levenshtein (LEVENSHTEIN, 1966). O sistema de classificação tem também um valor de semelhança mínimo, o qual é utilizado para identificar frases cujo o tipo não se encaixa em nenhuma das opções representadas no mapa.
Aplicando-se este algoritmo de classificação na frase apresentada no Quadro 9, por exemplo, ao submeter-se a frase ao Mecanismo de Atenção, este consultaria o dicionário de tipos de perguntas e verificaria que, para a identificação do tipo da pergunta, somente a palavra "quem" (linha 5) deve ser mantida na frase.

Quadro 9: Informações linguísticas geradas pelo parser ao analisar a frase "Quem foi a esposa do presidente americano Lincoln?"; em destaque a palavra utilizada para classificar o tipo da de pergunta.

```
1. SOURCE: Running text
2.  Quem foi a esposa do presidente americano Lincoln?
3.  A1
4.  QUE:fcl
5.  =Cs:spec("quem" <clb> <*> <interr> M/F S/P)  Quem
6.  =P:v-fin("ser" <fmc> <mv> PS 3S IND VFIN)   foi
7.  =S:np
8.  ==DN:art("o" <artd> F S)a
9.  ==H:n("esposa" <Hfam> F S)   esposa
10.  ==DN:pp
11.  ===H:prp("de" <sam-> <np-close>)        de
12.  ===DP:np
13.  ====DN:art("o" <-sam> <artd> M S)      o
14.  ====H:n("presidente" <Hprof> M S)      presidente
15.  ====DN:adj("americano" <nat> <jh> <np-close> M S)     americano
16.  ====DN:prop("Lincoln" <hum> <*> <np-close> M S)      Lincoln
17. =?
```

Após isto, o algoritmo de classificação usa a Distância de Levenshtein para identificar no mapa que a sequência do tipo QUEM (“quem <clb> <interr> M/F S/P” - linha 2 do Quadro

8) é a mais semelhante às anotações linguísticas da palavra mantida na frase pelo Mecanismo de Atenção (“quem <clb> <*> <interr> M/F S/P” - linha 5 do Quadro 9).

A Tabela 1 lista todos os tipos de perguntas atualmente identificáveis pelo sistema ENSEPRO. A estrutura de dados que representa os tipos de pergunta pode ser facilmente expandida. A inclusão de novos elementos é bastante simples e intuitiva, bastando adicionar as novas palavras no dicionário, se for o caso, e incluir as novas sequências na seção de mapeamento. Outro fator que favorece consideravelmente a inclusão de novos tipos é a legibilidade da estrutura de dados, uma vez que os nomes utilizados para representar as palavras são muito semelhantes às palavras da pergunta em si.

Tabela 1: Tipos de perguntas, respectivas sequências de palavras utilizadas para a sua caracterização linguística e exemplos de frases.

Tipo	**Sequências de Mapeamento**	**Exemplo se frases**
É_UM	é um	A proinsulina é uma proteína?
ALGUM	algum	Há algum jogo eletrônico chamado Battle Chess?
EM_QUE	em que	Em que estado dos EUA fica o Fort Knox?
QUANTO	quanto; quantas vezes	Quantas cadeiras têm o estádio do FC Porto?
QUANDO	quando; em que dia	Quando foi a Batalha de Gettysburg?
O_QUE	o que	O que é uma ontologia?
QUAL	qual; que	Qual papa fundou a Televisão do Vaticano?
QUEM	quem	Quem foram os pais da Rainha Victória?
ONDE	onde; aonde	Onde está enterrado Syngman Rhee?
CONSULTA	ache; liste; busque; encontre; descubra; localize; me dê; obtenha; recupere; consiga	Liste todos os membros da Prodigy. Ache as plataformas de lançamento da NASA. Mostre os netos de Elvis Presley.

Embora o principal objetivo da classificação dos tipos esteja relacionado ao processo de seleção de TRs, o tipo da questão é também usado pelo módulo CBC em alguns casos específicos. Se houver somente um TR na frase, é preciso recorrer-se ao tipo da questão para a localização da resposta na BC.

Este é o caso da frase apresentada para o tipo QUANDO da Tabela 1, por exemplo. A pergunta contém somente um TR (linha 7 do Quadro 10), o que impede a localização da resposta, pois é necessário ao menos duas informações para a localização de uma resposta no caso de BCs baseada em triplas.

Quadro 10: Exemplo de perguntas que contém somente um TR.

1. # ensepro -frase "Quando foi a Batalha de Gettysburg?" -tr -original
2. Analisando frase(s)...
3. -> Frase 1: Quando foi a Batalha de Gettysburg?
4. --> Tipo: TipoFrase={tipo=quando, **confianca=0.91**, ids=[2]}
5. --> Voz: Voz.ATIVA
6. --> Termos Relevantes:
7. **----> TR 0: [H:prop|6| Batalha_de_Gettysburg]**

Para o caso de perguntas como a apresentada no Quadro 10, é necessário que o módulo CBC considere o tipo da pergunta ao fazer a busca na BC, pois não é possível identificar a tripla que contém a resposta a partir de somente um TR. Um detalhamento mais aprofundado sobre esta questão é apresentado na seção 4.3.5 Seleção de respostas do módulo CBC.
Pode-se perceber que um nível de confiança é associado à classificação do tipo de pergunta (linha 4 do Quadro 10). A confiança da classificação é obtida pela normalização entre 0 e 1 da distância de Levenshtein entre as anotações linguísticas do dicionário e as palavras da frase selecionadas pelo Mecanismo de Atenção. O valor de confiança é o fator determinante para a classificação do tipo da pergunta.
Uma vez finalizada a classificação do tipo de pergunta, o próximo processo a ser realizado no módulo CLN é a identificação das palavras mais importantes da frase para a localização da resposta na BC, ou seja, a seleção dos termos relevantes.

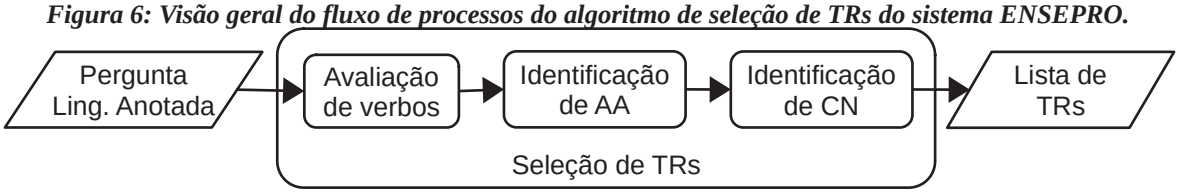
4.2.4 Seleção dos Termos Relevantes

Dentre os processos realizados na etapa de CLN, a seleção das palavras que serão utilizadas na busca da resposta na BC é um dos procedimentos mais importantes, pois a seleção dos termos relevantes (TR) para a consulta tem impacto direto na capacidade de localização da resposta do sistema ENSEPRO.

É importante ressaltar que optou-se por denominar de TR as palavras importantes da frase e não palavras chaves por que a seleção das palavras se dá com base em questões puramente linguísticas, sem uso de listas de palavras proibidas (stop words) ou índices estatísticos, abordagens geralmente utilizadas para a seleção de palavras chaves em sistemas de Recuperação de Informação.

O algoritmo implementado no sistema ENSEPRO avalia somente as informações de Part of Speech (POS) das palavras e determinados arranjos de termos específicos da frase no momento de selecionar os TRs. A Figura 6 apresenta uma visão geral do processo, o qual será detalhado nos próximos parágrafos.

Figura 6: Visão geral do fluxo de processos do algoritmo de seleção de TRs do sistema ENSEPRO.



Fonte: elaborada pelo autor.

O algoritmo de seleção dos TRs utiliza o tipo da pergunta para identificar o trecho da frase de onde serão buscados os termos que irão compor a consulta para localização das respostas na BC. O tipo da pergunta demarca o ponto de partida para o algoritmo de seleção de TRs. Somente podem ser TRs da frase os substantivos (comuns ou próprios) e os verbos encontrados após a(s) palavra(s) que determinam o tipo da pergunta.

Contudo, a seleção de TRs demanda uma busca seletiva que depende da estrutura frasal da pergunta, ou seja, do estilo de escrita de quem elaborou a frase. A detecção da presença de algumas estruturas linguísticas específicas na frase é imprescindível para a correta seleção dos TRs. A seleção dos TRs depende primeiramente da verificação da presença das estruturas linguísticas bem específicas como, por exemplo, as Locuções Verbais, os Adjuntos Adnominais e os Complementos Nominais.

4.2.4.1 Tratamento das Locuções Verbais

Como dito anteriormente, além dos substantivos, também os verbos são, via de regra, considerados como TRs para a localização da resposta à pergunta. No entanto, esta regra tem exceções que devem ser consideradas. Dependendo da forma pela qual o usuário expressou a sua pergunta, alguns verbos podem não assumir um papel relevante na busca da resposta, não devendo então ser considerados como TRs da frase.

A classificação de verbos quanto a sua relevância para a busca da resposta se dá pela identificação de locuções verbais na pergunta ou pela classificação semântica do verbo. A locução verbal (LV) é uma sequência verbos justapostos, iniciada por um ou mais verbos auxiliares conjugados e sempre finalizadas por um verbo principal no infinitivo, gerúndio ou particípio (BECHARA, 2012). Somente o verbo principal da locução verbal é considerado como TR.

Caso não seja identificado uma locução verbal na frase, identifica-se a relevância do verbo pela verificação da sua classe semântica, a qual se subdivide em dois grupos: relacionais e nocionais (BECHARA, 2012). O verbo somente será considerado TR quando pertencer à classe relacional. Os verbos nocionais são considerados como irrelevantes para a busca da resposta, pois têm significado muito amplo e vago (BECHARA, 2012).

No Quadro 11 apresenta-se um exemplo de pergunta contendo a LV “foi casada” (linha 7). Embora a frase contenha dois verbos, pode-se observar que somente o verbo principal “casada” é considerado pelo sistema como TR (linha 3), tendo sido ignorado o verbo auxiliar “foi”.

1. ensepro -lv -tr -original -frase "Quem **foi casada** com o presidente Chirac?"
2. --> Termos Relevantes:
3. **----> TR 0: [Vm:v-pcp|5| casada]**
4. ----> TR 1: [H:n|10| presidente]
5. ----> TR 2: [DN:prop|11| Chirac]
6. --> Locuções Verbais:
7. **----> LV 0: [foi, casada]**

***Quadro 11: Exemplo de pergunta em que a presença de locução verbal é detectada, implicando na escolha de somente um dos verbos como Termo Relevante*.**

Esta avaliação da relevância do verbo pela sua função dentro da LV é extremamente importante no momento de elaborar a consulta à BC, pois a inclusão de verbos auxiliares na consulta não contribuiria em nada para a localização da resposta.

Além da função do verbo dentro da LV, utiliza-se também a classe semântica dos verbos para avaliar a sua inserção na lista de TR. Caso a frase não contenha uma LV, define-se a relevância do verbo pela sua classe semântica, a qual se subdivide em dois grupos: verbos relacionais e nocionais.

Conforme definição dada por Bechara (2012), os verbos nocionais têm significado amplo e vago, por isto não sendo relevantes para a busca da resposta. Decorre daí que somente verbos relacionais podem ser considerados como TR.

Na prática, implementou-se um sistema bastante simples para a avaliação da classe semântica dos verbos: uma lista de verbos nocionais. Caso o verbo encontrado na pergunta não esteja relacionado na lista de verbos nocionais, ele é então considerado como um verbo relacional e consequentemente inserido na lista de TR. Caso contrário, é ignorado. No Quadro 12 pode-se ver um exemplo de pergunta contendo o verbo “foi”, o qual não é considerado como TR por ser um verbo nocional.

1. ensepro -tr -original -frase "Quando **foi** a Batalha de Gettysburg?"
2. --> Termos Relevantes:
3. ----> TR 0: [H:prop|6| Batalha_de_Gettysburg]

***Quadro 12: Exemplo de pergunta em que a avaliação da relevância do verbo se dá pela verificação da sua classe semântica*.**

É importante ressaltar que ambos os procedimentos de avaliação da relevância de verbos são derivados exclusivamente das informações sintáticas e estruturais da frase fornecidas pelo parser linguístico. Em relação a avaliação de verbos, a combinação destes dois procedimentos bastante simples, mas altamente precisos, são suficientes para a classificação de todo e qualquer verbo que as perguntas venham a ter. Nas próximas seções são vistos procedimentos para auxiliar na avaliação da relevância da outra classe gramatical passível de ser considerada TR: os substantivos.

4.2.4.2 Tratamento dos Adjuntos Adnominais

É importante esclarecer inicialmente que, embora tenha sido adotado o vocabulário linguístico para denominar alguns arranjos específicos de substantivos, o conceito de Adjunto Adnominal (AA) no contexto deste trabalho não é idêntico ao conceito original da Linguística.

Deu-se aqui o nome de AA a toda e qualquer sequência nominal (substantivos ou adjetivos) obrigatoriamente justaposta e consecutiva. Esta formação obrigatória não se aplica à função sintática homônima da Linguística, para a qual o AA é caracterizado como uma sequência consecutiva de nomes, mas não obrigatoriamente justapostos.

Quando o AA contém um substantivo próprio, pode-se supor que há uma relação taxonômica entre os elementos que constituem o AA. Na pergunta apresentada no Quadro 13, por exemplo, o sistema ENSEPRO identifica a presença de dois AA: presidente americano (linha 5) e presidente Lincoln (linha 6).

1. # ensepro **-aa** -frase "Quem foi a esposa do presidente americano Lincoln?"
2. Analisando frase(s)...
3. -> Frase 1: Quem foi a esposa do presidente americano Lincoln?
4. --> Adjuntos Nominais:
5. **----> AA 1: presidente + americano**
6. **----> AA 2: presidente + Lincoln**

Quadro 13: Adjuntos Adnominais identificados pelo sistema ENSEPRO.

Os AA detectados na frase do Quadro 13 permitem concluir que Lincoln é um presidente e que este presidente (Lincoln) é americano. Podemos então supor que as triplas que contem a resposta correta à pergunta deverão confirmar a relação [Lincoln é_um presidente] e [Lincoln é_um americano].

A informação de que existe a relação taxonômica (é_um) entre os componentes do AA é uma informação importante, pois possibilita deduzir/predizer os três elementos de triplas que, combinadas com outras triplas, ajudarão a localizar a resposta correta para a questão.

Se por um lado é possível usar o AA como um aliado na localização da resposta correta, por outro lado verificou-se em experimentos que, em BC contendo informações incompletas (como por exemplo a DBPedia), temos uma inversão de função: o AA vai indicar elementos da frase que, mesmo sendo TRs, não devem ser considerados como essenciais para a localização da resposta.

A inversão de função do AA ocorre por conta da falta de informações da BC. Em princípio, os indivíduos de uma BC devem ter o seu tipo definido. Na frase do Quadro 13, por exemplo, conforme já comentado anteriormente, sabe-se que Lincoln, no contexto da BC, é um indivíduo, pois é um substantivo próprio. Sendo um indivíduo, supõe-se que encontraremos na BC uma ou mais triplas que vão definir o tipo deste indivíduo ([Lincoln é_uma Pessoa] por exemplo).

Conforme já comentado, os AA encontrados na questão nos permitem supor que deveríamos encontrar triplas na BC que confirmem que Lincoln é um presidente e é americano. No entanto, estas triplas não existem na DBPedia. Neste caso então, o AA passou

a indicar os termos da frase que, mesmo sendo considerados TR devido à classe gramatical que pertencem, podem ser ignorados para a localização da resposta.

Mais que ignorados, no caso de utilizar-se uma BC incompleta com o sistema ENSEPRO, as palavras integrantes de AA podem ser prejudiciais para a localização da resposta. Se considerar-se os integrantes do AA como TR na localização da resposta (ou seja, incluí-los como palavras chaves na consulta), obtém-se uma grande quantidade de triplas sem relação alguma com a resposta para a pergunta, aumentando-se consideravelmente o nível de ruído, prejudicando-se assim o algoritmo de seleção de respostas.

É por isto que para o sistema ENSEPRO, independentemente da classe gramatical (substantivos comuns ou adjetivos), os integrantes do AA não serão considerados como TR. Como se pode ver na linha 1 do Quadro 14, levando-se em conta somente a classe gramatical das palavras que compõem a frase, o sistema ENSEPRO identificou 3 TRs: os substantivos comuns esposa (linha 7) e presidente (linha 8) e o substantivo próprio Lincoln (linha 9).

1. # ensepro **-tr -original** -frase "Quem foi a esposa do presidente americano Lincoln?"
2. Analisando frase(s)...
3. -> Frase 1: Quem foi a esposa do presidente americano Lincoln?
4. --> Tipo: TipoFrase={tipo=quem, confianca=0.89, ids=[2]}
5. --> Voz: Voz.ATIVA
6. --> Termos Relevantes:
7. **----> TR 0: [H:n|6| esposa]**
8. **----> TR 1: [H:n|11| presidente]**
9. **----> TR 2: [DN:prop|13| Lincoln**]

10. # ensepro **-tr -final** -frase "Quem foi a esposa do presidente americano Lincoln?"
11. Analisando frase(s)...
12. -> Frase 3: Quem foi a esposa de o <split> Abraham_lincoln <split>
13. --> Tipo: TipoFrase={tipo=quem, confianca=0.69, ids=[2]}
14. --> Voz: Voz.ATIVA
15. --> Termos Relevantes:
16. **----> TR 0: [H:n|6| esposa]**
17. **----> TR 1: [H:prop|11| Abraham_lincoln]**

Quadro 14: listagem dos TRs antes e depois da detecção da presença de um AA na frase.

Após a detecção de presença do AA na frase, o substantivo comum presidente, por ser integrante de um AA, foi removido da lista final de TRs (linhas 16 e 17). É importante ressaltar novamente que a remoção de TRs do tipo AA só é benéfica para localização da resposta no caso de BC incompletas.

Quando a BC define de forma mais rigorosa os tipos dos indivíduos, o AA auxilia na identificação das triplas que vão compor a resposta correta. Uma vez que se tem como objetivo que o sistema ENSEPRO seja um sistema genérico de PRS, então decidiu-se optar pela remoção dos AA da lista de TRs.

Contudo, implementou-se como um parâmetro configurável do sistema a opção de considerar-se ou não os componentes do AA como TR, sendo então possível adequar o comportamento do sistema ENSEPRO ao contexto da BC a ser utilizada.

4.2.4.3 Tratamento dos Complementos Nominais

Toma-se novamente emprestado da Linguística o termo Complemento Nominal (CN) para designar uma sequência de nomes (substantivos ou adjetivos) unidos por preposições. Também como o AA, o conceito de CN adotado neste trabalho é ligeiramente diferente do seu homônimo na linguística, uma vez que aqui toda e qualquer sequência de nomes unidos por preposições são considerados CN, enquanto que na Linguística podem haver sequências de nomes que são AA.

Embora a presença de CNs seja utilizada somente no módulo de CBC, por ser este um processo que envolve a análise das anotações linguísticas e da estrutura da frase, implementou-se a sua detecção no módulo CLN.

A identificação dos CNs ocorre pela busca na frase de sequências de nomes (substantivos ou adjetivos) separados por preposições. No exemplo apresentado no Quadro 15, pode-se ver que foram identificados dois CNs: “prefeito da capital” (linha 5) e “capital da Polinésia Francesa” (linha 6).

```
1. # ensepro -cn -frase "Quem é o prefeito da capital da Polinésia Francesa?"
2. Analisando frase(s)...
3. -> Frase 1: Quem foi a esposa do presidente americano Lincoln?
4. --> Complementos Nominais:
5. ----> CN 1: prefeito + capital
6. ----> CN 2: capital + Polinésia_Francesa
```

Quadro 15: Complementos nominais identificados pelo sistema ENSEPRO.

A presença de CNs na pergunta podem indicar relações semânticas específicas entre os TRs da frase, as quais podem ser utilizadas para a realização de procedimentos adicionais no momento da elaboração da consulta.

A função do módulo CLN é somente buscar e identificar os CNs, uma vez que a sua análise e possível uso são vistos somente no módulo CBC (mais detalhes sobre o uso dos CNs podem ser vistos na seção 4.3.2 Injeção de TRs por subconsulta).

4.2.5 Expansão dos Termos Relevantes

O processo de seleção dos TRs da frase tem como objetivo identificar as palavras chaves para a busca das resposta na BC. Contudo, existe a possibilidade de que os termos escolhidos pelo usuário ao formular a pergunta não sejam os mesmos termos utilizados ao elaborar o vocabulário da BC.
É necessário encontrar uma forma de contornar esta diferença de vocabulário entre a frase e a BC. Geralmente busca-se a solução para este problema na utilização de algum mecanismo que permita expandir a lista de palavras pela inserção de outros termos com significado semelhante.

No sistema ENSEPRO foram implementados dois mecanismos para expansão dos termos: sinônimos via Wordnet e nominalização de verbos. Ambos os processos de expansão da lista de termos relevantes são detalhados nas próximas seções.

4.2.5.1 Seleção de Sinônimos via Wordnet

O uso dos synsets da Wordnet para expandir as palavras chaves é uma abordagem bastante utilizada em sistemas de RI e PR, não sendo diferente para os sistemas de PRS. A abordagem utilizada no sistema ENSEPRO consiste em utilizar a forma canônica da palavra (informação fornecida pelo parser linguístico) para localizar os sinônimos na Wordnet.

O procedimento de expansão dos TRs se dá pela inclusão dos seus sinônimos através de uma consulta aos synsets da Open Multilingual Wordnet[20] (OMW) via interface disponibilizada no NLTK[21]. No Quadro 16 apresenta-se um exemplo do retorno da consulta à OMW para expansão dos TRs proinsulina e proteína.

1. # ensepro **-sin** -tr -original -frase "A proinsulina é uma proteína?"
2. --> Termos Relevantes:
3. **----> TR 0: [H:n|4| proinsulina]**
4. **--------> Sinonimos: {'por': [], 'eng': []}**
5. **----> TR 1: [H:n|8| proteína]**
6. **--------> Sinonimos: {'por': [proteínas], 'eng': [protein]}**

Quadro 16: Lista de TRs e respectivos sinônimos em inglês e português.

Pode-se observar que não há sinônimos para proinsulina na OMW, o que resulta em uma lista de sinônimos vazia (linha 4 do Quadro 16). Isto ocorre porque não está cadastrado na OMW um sinônimo para o termo “proinsulina”. Para o TR “proteína”, no entanto, foram encontrados os sinônimos “proteínas” em português e “protein” em inglês. A partir deste retorno, o sistema ENSEPRO incluirá os sinônimos em português e inglês na lista de TRs a serem consultadas na BC.

O ENSEPRO usa os sinônimos em inglês porque é extremamente comum a adoção do inglês como vocabulário para as BCs. Isto significa que o sistema está pronto para buscar respostas em BCs elaboradas em português ou inglês. Embora não tenha sido realizado nenhum experimento, da mesma forma como foi implementado o suporte ao inglês, seria possível incluir no sistema o suporte a quaisquer uma das 25 línguas atualmente suportadas pela OMW.

Outra decisão importante em relação ao uso da OMW para expansão da lista de TRs foi restringir a inclusão dos sinônimos de classe gramatical idêntica à da palavra original da frase. Ao realizar a busca dos sinônimos a partir somente da palavra, sem o contexto, são retornados os sinônimos para todas as acepções conhecidas do termo. É necessário evitar esta situação, pois os sinônimos de acepções diferentes daquela utilizada na frase podem prejudicar a busca da resposta.

[20] http://compling.hss.ntu.edu.sg/omw/
[21] http://www.nltk.org/

A análise de um exemplo torna mais clara a questão. Pode-se observar a partir da linha 5 do Quadro 17 que o sistema ENSEPRO identificou escalar e Monte Everest como TRs da frase. A palavra escalar tem duas acepções possíveis: uma em que expressa a ação de subir em algo, quando então é um verbo, e outra quando define características matemáticas de um número, quando então cumpre com o papel de adjetivo.

```
1. # python
2. >>> from nltk.corpus import wordnet as wn
3. >>> wn.lemmas('escalar', lang='por')
4. [Lemma('escalade.v.01.escalar'), Lemma('scalar.a.02.escalar')]

5. # ensepro -sin -tr -original -frase "Quem foi o primeiro a escalar o Monte Everest?"
6. --> Termos Relevantes:
7. ----> TR 0: [P:v-inf|10| escalar]
8. --------> Sinonimos: {'por': [escalar], 'eng': [escalade]}
9. ----> TR 1: [H:prop|13| Monte_Everest]
10. --------> Sinonimos: {'por': [], 'eng': []}
```

Quadro 17: Lista de sinônimos para a palavra “escalar” na OMW e no sistema ENSEPRO.

Pode-se perceber que a palavra escalar nesta frase está relacionada à ação de subir, uma vez que é o verbo da frase (v-inf – linha 7 do Quadro 17). O parser linguístico usa o contexto da frase para definir a correta classificação gramatical da palavra e, consequentemente, o significado mais apropriado.

Ao consultar-se a OMW (linha 3), são retornados os sinônimos para as duas acepções da palavra (linha 4): verbo (escalade.v.01.escalar) e adjetivo (scalar.a.02.escalar). O sistema ENSEPRO usa a informação fornecida pelo parser linguístico para selecionar somente os sinônimos da acepção relacionada à classe gramatical da palavra (linha 8).

Com o procedimento descrito acima pretende-se que os sinônimos inseridos na lista de TRs estejam relacionados ao significado da palavra no contexto da frase. Finalizada a etapa de inserção de sinônimos, tem início o procedimento de expansão da lista de TRs via nominalização de verbos.

4.2.5.2 Nominalização dos Verbos

O objetivo da etapa de expansão da lista de TRs é gerar variações lexicais das palavras chaves da frase, visando aumentar a probabilidade de encontrar a resposta a partir dos termos fornecidos na pergunta. Conforme visto na seção 4.2.4 Seleção dos Termos Relevantes, os verbos com função semântica relacional são palavras chaves da pergunta para a localização da resposta.

Viu-se também naquela seção que os verbos relacionais via de regra cumprirão com o papel de predicados nas triplas, dada a sua função semântica de expressar relações. Se por um lado sabe-se de antemão a posição do verbo nas triplas da BC, por outro lado é pouco usual utilizar a sua forma verbal, sendo mais comum a utilização da sua forma nominal.

A situação descrita acima é exemplificada no Quadro 18, o qual apresenta uma pergunta contendo dois TR: o verbo criou (linha 3) e o substantivo próprio Batman (linha 4). Estes dois TR serão utilizados para a busca das respostas na BC (linha 5).

1. # ensepro -tr -original -frase "Quem criou Batman?"
2. --> Termos Relevantes:
3. **----> TR 0: [P:v-fin|3| criou]**
4. ----> TR 1: [Od:prop|4| Batman]

5. Respostas:
6. **['batman', 'criador', 'bill_finger']**
7. **['batman', 'criador', 'bob_kane']**

8. Formas nominais do verbo **criar** no NomLex-PT:
9. - criação, criada, **criador**, criatura

Quadro 18: Pergunta contendo TR verbal cujas respostas na BC estão representadas por triplas que tem como predicado a forma nominal do verbo no NomLex-PT.

Além da pergunta, apresenta-se também no Quadro 18 as respostas (linhas 6 e 7). Pode-se observar que o substantivo próprio da pergunta (Batman – linha 4) é lexicalmente igual ao sujeito de ambas as triplas que respondem a questão. No entanto, o verbo da pergunta (criou – linha 3) é lexicalmente representando nas triplas em uma das suas possíveis formas nominais (criador – linha 9).

Verificou-se empiricamente que a opção de representar lexicalmente as relações pela forma nominal é uma abordagem bastante usual. Diante desta constatação, decidiu-se que, para os TR verbais, além dos sinônimos, seriam incluídos também na lista as suas formas nominais.

Para a geração das formas nominais dos verbos fez-se uso do NomLex-PT (DE PAIVA et al., 2014), uma lista de verbos em português e respectivas formas nominais disponibilizada publicamente no formato RDF[22]. Sempre que uma pergunta contiver um verbo relacional, o sistema busca os seus sinônimos e, para cada sinônimo, as suas respectivas formas nominais, acrescentando-os na lista de TR.

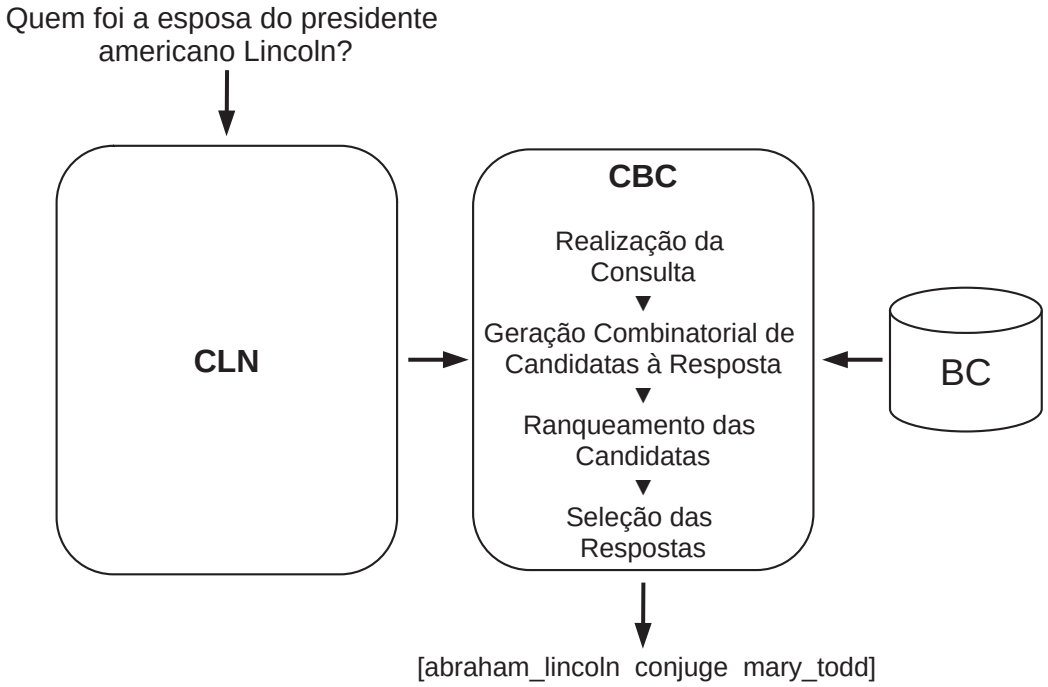
A expansão da lista de TR pela inserção das formas nominais dos verbos é o último processo realizado no módulo CLN. Todas as informações linguísticas necessárias para a realização da consulta à BC estão prontas para serem utilizadas pelo módulo CBC.

4.3 A Consulta à Base de Conhecimento

O módulo CBC tem por objetivo gerar e realizar a consulta à BC, bem como selecionar as respostas que serão retornadas pelo sistema. Como pode ser visto na Figura 7, para realizar esta tarefa o módulo CBC executa a seguinte sequência de processos: Realização da Consulta, Geração Combinatorial de Candidatos à Resposta, Ranqueamento de Candidatos e Seleção de Respostas.

[22] https://github.com/arademaker/openWordnet-PT

Figura 7: Visão geral do fluxo de processos do módulo CBC.



Fonte: elaborada pelo autor.

O primeiro processo do módulo CBC é a elaboração e execução de uma consulta para buscar as triplas candidatas à resposta. Estas triplas são combinadas seguindo algumas regras de formação e ordenadas segundo um critério de ranqueamento, o qual possibilitará a realização da seleção da resposta.

O resultado obtido ao final da execução do módulo CBC é o conjunto de triplas que respondem à pergunta recebida pelo sistema ENSEPRO. Caso o sistema não encontre na BC a resposta, o retorno será uma lista vazia. Nas próximas seções serão vistos em detalhes cada um destes processos.

4.3.1 Consulta à Base de Conhecimento

Um pressuposto da abordagem implementada neste trabalho parte do princípio de que o processo de localização da resposta para uma pergunta em LN pode beneficiar-se das informações linguísticas da frase. Viu-se nas seções anteriores que as informações linguísticas coletadas no início do módulo CLN são utilizadas para realizar vários processos do sistema ENSEPRO.

O módulo CBC também faz uso das informações linguísticas em vários dos seus subprocessos, como na elaboração da consulta que será executada para localizar na BC as triplas que respondem à questão recebida pelo sistema. Esta consulta tem como objetivo principal criar uma lista de triplas candidatas à resposta a partir da lista de TRs.

A elaboração da consulta à BC inicia pela inserção dos elementos da lista de TRs em uma consulta, a qual será submetida ao Elasticsearch para execução (para mais detalhes, veja a seção 5.1.1 Pré-processamento da Base de Conhecimento). Esta inserção é realizada com base nas informações linguísticas dos TRs.

Conforme já visto na seção 4.2.4 Seleção dos Termos Relevantes, a lista de TRs é composta não somente por palavras originadas da frase, mas também por TRs derivados, os quais foram incluídos na lista por serem sinônimos ou nominalizações dos TRs originais.

Todos os TRs (originais e derivados) são usados na consulta à BC que vai popular a lista de triplas candidatas à resposta. O que define como os TRs serão inseridos nesta consulta é a sua classe gramatical.

Caso o TR seja um verbo, por exemplo, sabe-se que ele é um verbo relacional, pois somente verbos relacionais podem ser TRs. Os verbos relacionais são aqueles verbos que, como o próprio nome diz, estabelecem um relacionamento entre os elementos da frase. No âmbito de uma BC, os relacionamentos são representados (sempre e somente) no predicado da tripla.

Podemos concluir então que um TR verbal, bem como as suas derivações, somente podem ocorrer em uma BC como predicado de tripla. A partir da identificação de um TR verbal na frase, o sistema ENSEPRO vai buscar na BC e inserir na lista de candidatas à resposta todas as triplas cujo o campo predicado case total ou parcialmente com o TR verbal original ou com suas derivações (sinônimos e nominalizações).

Um raciocínio bastante semelhante é usado no caso do TR ser um substantivo próprio. O substantivo próprio tem como função na frase referenciar um indivíduo ou entidade do mundo real. Em relação a uma BC semântica, os indivíduos do mundo real são sempre ou sujeitos ou objetos da tripla.

A partir disto, pode-se concluir que um TR originado de um substantivo próprio somente poderá aparecer como sujeito ou como objeto de uma tripla. Assim sendo, o módulo CBC vai buscar todas as triplas da BC cujos campos sujeito ou objeto casem total ou parcialmente com os TRs que são substantivos próprios e inseri-las também na lista de triplas candidatas à resposta.

Por fim, caso o TR seja um substantivo comum, não há nenhuma previsão quanto ao papel deste na tripla, pois o substantivo comum pode ser sujeito, predicado ou objeto. Por isto o módulo CBC busca e insere na lista de candidatas todas as triplas que casem total ou parcialmente com os TRs que são substantivos próprios em qualquer uma das posições das triplas.

A análise de um exemplo prático vai ajudar a compreender os procedimentos descritos nos parágrafos anteriores. No Quadro 19 pode-se ver os TRs originais da frase e seus derivados, todos identificados no decorrer da execução do módulo CLN (linhas 1 a 9), e a respectiva consulta (linhas 10 a 19) que será executada para buscar na BC as triplas candidatas à resposta.

```
1.  root@ensepro:~# ensepro -tr -final -sin -frase "Que rio a Ponte do Brooklyn cruza?"
2.  --> Termos Relevantes:
3.  ----> TR 0: [H:n|4| rio]
4.  --------> Sinonimos: {'por': [rio], 'eng': [river]}
5.  ----> TR 1: [H:prop|7| Ponte_do_Brooklyn]
6.  --------> Sinonimos: {'por': [], 'eng': []}
7.  ----> TR 2: [P:v-fin|8| cruza]
8.  --------> Sinonimos: {'por': cruzar, atravessar], 'eng': [cross, hybridise, cruise]}
9.  --------> Nominalizacoes: {'cruzador', 'cruzamento'}

10. Consultas geradas:
11. "query": {
12.    "bool": {
13.       "should": [
14.          { "terms": { "sujeito.ngram_conceito": [ "rio", "river", "ponte_do_brooklyn"
    ] } },
15.          { "terms": { "predicado.ngram_conceito": [ "rio", "river", "cruzar",
    "atravessar",                    "cross", "cruise", "cruzador", "cruzamento"  ] } },
16.          { "terms": { "objeto.ngram_conceito": [ "rio", "river",
                    "ponte_do_brooklyn" ] } }
17.       ]
18.    }
19. }
```

Quadro 19: Exemplo de consulta gerada a partir dos TRs de uma pergunta.

Pode-se observar no Quadro 19 que o TR verbal cruza e seus derivados (linhas 7 a 9) aparecem somente no campo predicado da consulta (linha 15), enquanto que o TR substantivo próprio Ponte_do_Brooklyn (linha 5) aparece nos campos sujeito (linha 14) e objeto (linha 16). Já o TR substantivo comum rio e seu derivado (linha 3 e 4) aparecem em todos os três campos da consulta.

O retorno da execução da consulta é uma lista com todas as triplas da BC que casam total ou parcialmente com ao menos um dos TRs da pergunta, referenciada como lista de triplas candidatas à resposta. Dependendo dos termos utilizados para expressar a pergunta, a lista de triplas candidatas pode ser bastante longa. Este é, aliás, um dos benefícios de utilizar-se as informações linguísticas dos TRs para elaborar a consulta e selecionar o menor número de triplas possível.

Se, por um lado, dificilmente haverá na BC uma tripla com o predicado contendo a palavra Ponte_do_Brooklyn, por outro lado há uma grande probabilidade de haver um número significativo de triplas contendo o TR verbal e seus derivados nos campos sujeito ou objeto da tripla.

O impacto do uso da classe gramatical do TR fica bastante evidente ao analisar-se um caso de uso prático. Ao buscar a lista de candidatas à resposta para a pergunta do Quadro 19 na DBPedia em português (versão 2016-10) utilizando-se as classes gramaticais dos TRs, obtém-se como resultado uma lista com 322.359 triplas candidatas.

Realizando-se exatamente a mesma consulta, mas desta vez desconsiderando-se a classe gramatical dos TRs, ou seja, buscando-se todos os TRs nos três campos da tripla (sujeito, predicado e objeto), obtém-se como resultado em uma lista de candidatas com 388.051 triplas, ou seja, uma lista 20,37% maior do que a lista gerada com o uso das informações linguísticas

Contudo, o principal objetivo de usar a classe gramatical do TR no momento de elaborar a lista de triplas candidatas não é somente diminuir a quantidade de triplas. O principal benefício obtido é melhorar a qualidade dos dados inseridos na lista de candidatas, pois usando-se as informações sintáticas dos TRs diminui-se a inserção de triplas com informações não relacionadas à pergunta.

Além do impacto significativo na quantidade e qualidade dos dados da lista de candidatas, a redução do tamanho da lista de candidatas também diminui o tempo de resposta do sistema, pois resulta em uma expressiva economia do esforço computacional do sistema ENSEPRO nas próximas etapas do processo.

Embora o processo de consulta à BC descrito nesta seção se aplique a uma grande parte das perguntas, existe um tipo de pergunta específica que demanda um procedimento inicial antes de seguirem-se os passos descritos acima. A próxima seção descreve os detalhes relativos a este caso específico de pergunta.

4.3.2 Injeção de TRs por subconsulta

A lista de TRs é originalmente produzida a partir das palavras que compõem a pergunta. A partir da lista de TRs é elaborada a consulta que vai gerar a lista de triplas candidatas a resposta. Pode-se ver que existe uma relação direta entre a lista de TRs e a composição da lista de candidatas.

No entanto, dependendo de como a pergunta foi formulada, pode ocorrer uma situação em que um TR necessário para localização da resposta não esteja explícito na frase. Para que o módulo CBC possa gerar uma lista de candidatas que efetivamente contenha a resposta à pergunta é obrigatório primeiramente encontrar este TR subentendido.

Diz-se que este TR é subentendido porque ele não está explícito na pergunta, mas sua presença é pressuposta (subentendida) pela referência. No caso da pergunta apresentada no Quadro 20, por exemplo, não será possível ao módulo CBC gerar a lista de triplas candidatas que contenha a resposta correta utilizando-se somente dos TRs prefeito, capital e Polinésia Francesa (linhas 3 a 5) localizados na frase da pergunta.

```
1. # ensepro -tr -final -cn -frase "Quem é o prefeito da capital da Polinésia Francesa?"
2. --> Termos Relevantes:
3. ----> TR 0: [H:n|6| prefeito]
4. ----> TR 1: [H:n|11| capital]
5. ----> TR 2: [H:prop|16| Polinésia_Francesa]
6. --> Complementos Nominais:
7. ----> CN 0: prefeito + capital
8. ----> CN 1: capital + Polinésia_Francesa
```

Quadro 20: Exemplo de pergunta contendo TR oculto a ser descoberto por subconsulta.

Antes de gerar a lista de candidatas é necessário primeiramente identificar se a pergunta é do tipo que contém TR subentendido. Esta identificação se dá pela análise da presença de Complementos Nominais (CN) que a pergunta possa ter.

A primeira atividade realizada pelo módulo CBC é identificar a existência na frase de sequências justapostas de CNs interseccionados entre si. A presença desta estrutura específica de CNs indica que a frase contém um TR subentendido.

A frase apresentada no Quadro 20 é um exemplo de pergunta que contém dois CNs justapostos correspondendo aos trechos “prefeito da capital” (linha 7) e “capital da Polinésia Francesa” (linha 8). Estes CNs estão interseccionados pelo compartilhamento da palavra “capital”.

A partir desta constatação, sabe-se que a pergunta contém um TR subentendido, o qual precisa ser localizado na BC para ser inserido na lista de TRs da frase. A localização do TR subentendido é realizada através de uma subconsulta[23] à BC, a qual é elaborada/construída a partir de um subconjunto específico dos TRs da frase.

Em relação ao algoritmo para identificação do CN que vai dar origem à subconsulta, este é baseado na busca do CN que pode ser resolvido a partir dos termos explícitos na pergunta, referido como CN resolvível. O CN resolvível é aquele CN cujos TRs possibilitam a localização do TR subentendido, pois contém somete referências explícitas a indivíduos do mundo real, ou seja, um substantivo próprio

Analisando a pergunta do Quadro 20, verifica-se que os TRs prefeito e capital do CN 0 (linha 7) não representam um indivíduo específico, pois são ambos substantivos comuns representando classes de indivíduos. Em contrapartida, verifica-se que o CN 1 (linha 8) apresenta sim um TR associado a um indivíduo específico: o substantivo próprio Polinésia Francesa, sendo portanto este o CN resolvível da frase.

Identificado o CN resolvível, passa-se à elaboração da subconsulta para buscar na BC o TR subentendido. A subconsulta é elaborada de forma a recuperar da BC todas as triplas que contenham o substantivo próprio do CN resolvível como sujeito ou predicado e o substantivo comum do CN resolvível como predicado.

No Quadro 21 pode-se observar a subconsulta gerada para localizar o TR subentendido da pergunta “Quem é o prefeito da capital da Polinésia Francesa?”. O resultado retornado pela subconsulta (linha 17) contém o TR subentendido procurado: a palavra “papeete”. Identificado o TR subentendido, este é então inserido na lista de TRs.

---

[23] Utiliza-se o termo “*subconsulta*” para diferenciar esta consulta feita para localização do TR subentendido daquela descrita na seção 4.3.1 Consulta à Base de Conhecimento.

```
1.  {
2.      "query": {
3.        "bool": {
4.           "must": [ {
5.              "bool": {
6.                 "should": [
7.                    { "terms": { "sujeito.ngram_conceito": [ "polinesia_francesa" ] } },
8.                    { "terms": { "objeto.ngram_conceito": [ "polinesia_francesa"  ] } }
9.                    ]
10.                }
11.             },
12.             { "terms": { "predicado.ngram_conceito": [ "capital" ] } }
13.          ]
14.       }
15.   }
16. }

17. Retorno da subconsulta: [ *polinesia_francesa | *capital | papeete]
```

Quadro 21: *Subconsulta gerada a partir dos TRs do CN resolvível da pergunta "Quem é o prefeito da capital da Polinésia Francesa?"*.

Uma vez inserido o termo subentendido na lista de TRs da frase, o algoritmo de consulta segue o processo descrito na seção 4.3.1 Consulta à Base de Conhecimento, inserindo os demais TRs da frase para então gerar a lista de triplas candidatas a resposta. A lista de candidatas é então submetida ao processo de geração combinatorial de candidatas, processo descrito na próxima seção.

4.3.3 Geração combinatorial de candidatas à resposta

O resultado da execução da consulta à BC é uma lista, geralmente bastante extensa, contendo todas as triplas candidatas à resposta. Caso a BC contenha a resposta para a questão recebida pelo sistema ENSEPRO, então esta lista de candidatas a contém.

Contudo, supondo que a BC contenha a resposta para a questão, pode acontecer que a resposta completa para a questão esteja representada não em somente uma tripla, mas sim em uma combinação de triplas. O Quadro 22 apresenta um exemplo de pergunta (linha 1) em que a resposta (linha 3) está representada em um par de triplas da BC.

```
1. # ensepro -resposta -frase "Quem é o prefeito da capital da Polinésia Francesa?"
2. --> Melhores respostas:
3. ['*polinesia_francesa',  '*capital', '*papeete'] ['*papeete', '*prefeito',
   'michel_buillard']
```

Quadro 22: Exemplo de pergunta em que a melhor resposta é um par de triplas.

A lista de candidatas geradas a partir do processo de consulta à BC contém individualmente todas as triplas da resposta. Para que o sistema ENSEPRO possa retornar a

resposta correta para uma questão como a do Quadro 22, por exemplo, é necessário que as triplas individuais da lista de candidatas sejam combinadas em pares.

A geração dos pares de triplas é realizada por um algoritmo que varre a lista de candidatas combinando as triplas em pares seguindo um conjunto de regras predefinidas. Então, para cada tripla da lista de candidatas verifica-se a possibilidade de combiná-la com todas as demais triplas da lista. O pseudocódigo deste algoritmo é apresentado no Quadro 23.

```
1. Para tripla1 em Lista_de_Candidatas
2.     Para tripla2 em Lista_de_Candidatas
3.         Se tripla1 != tripla2
4.             Se tripla1.sujeito == tripla2.sujeito
5.                 Lista_de_Pares.adiciona(tripla1,tripla2)
6.             Se tripla1.sujeito == tripla2.objeto
7.                 Lista_de_Pares.adiciona(tripla1,tripla2)
```

***Quadro 23: Pseudo código do algoritmo de geração combinatorial de pares de triplas*.**

Como se pode ver, o código é bastante compacto, envolvendo operações simples de comparação e inserção na lista. Contudo, complexidade deste algoritmo é O(n!), estabelecendo uma relação direta entre a quantidade de triplas da lista de candidatas e o número de execuções do algoritmo de combinação de triplas.

Utilizando-se o sistema ENSEPRO em BC extensas, como a DBPedia em Português por exemplo, a lista de candidatas pode conter centenas de milhares de triplas, resultando em um número extremamente alto de execuções do código de geração de combinações.

Neste contexto, o uso das informações gramaticais no momento da elaboração da consulta à BC vista na seção 4.3.1 Consulta à Base de Conhecimento, ganha ainda mais importância, dada a redução significativa da lista proporcionada pelo procedimento. Contudo, mostrou-se necessária a implementação de mais otimizações no processo de geração combinatorial de candidatas.

Dado a complexidade O(n!) do algoritmo, tornou-se imprescindível implementar-se a paralelização da geração de combinações. Esta paralelização foi primeiramente implementada na linguagem Python, por ser esta a linguagem de programação escolhida para a implementação do sistema ENSEPRO. Contudo, o desempenho da linguagem Python não se mostrou adequado ao contexto.

É por este motivo que o código do gerador de combinações foi implementado em Java, uma vez que somente nesta linguagem de programação o desempenho da execução paralelizada obteve o desempenho esperado.

Outra decisão relacionada a complexidade do algoritmo de geração de combinações foi a limitação de geração de combinações a somente pares de triplas. A geração de trincas de triplas foi implementada, contudo o tempo de geração de combinações em trincas de triplas mostrou-se incompatível com o tempo de resposta esperado para um sistema de PR.

A partir desta definição, limitou-se atualmente a capacidade de resolução de perguntas do sistema ENSEPRO a respostas representadas em uma ou duas triplas. Perguntas cuja a resposta dependam de três ou mais triplas não são suportadas na versão atual do sistema.

A lista de pares de candidatas geradas é combinada com a lista de candidatas individuais (gerada na etapa da consulta à BC), produzindo uma única lista com todas as triplas candidatas à resposta. Esta lista de candidatas é submetida ao processo de ranqueamento, assunto visto na próxima seção.

4.3.4 Ranqueamento de respostas

Em princípio, se existir na BC uma resposta para a questão recebida pelo sistema, ela estará na lista de candidatas. É do módulo CBC a tarefa de localizar a resposta correta entre as candidatas. O processo de ranqueamento das candidatas analisa cada uma das triplas e calcula um escore de ranqueamento baseado em três métricas de avaliação.

As métricas levam em conta diversos atributos para avaliar o quanto cada tripla candidata tem possibilidade de ser a resposta para a questão recebida pelo sistema. As três métricas são somadas, gerando um valor único de ranque, o qual será utilizado para localizar a resposta entre as candidatas.

No Quadro 24 pode-se ver o acionamento do sistema ENSEPRO com as opções de retornar os TRs da frase, a melhor resposta dentre as candidatas (neste caso, a resposta correta para pergunta) e abaixo a lista com as duas primeiras triplas da lista de candidatas[24].

```
1.  # ensepro  -tr -original -resposta -frase "Que rio a Ponte do Brooklyn cruza?"
2.  --> Termos Relevantes:
3.  ----> TR 0: [H:n|4| rio]
4.  ----> TR 1: [H:prop|7| Ponte_do_Brooklyn]
5.  ----> TR 2: [P:v-fin|8| cruza]
6.  Melhores respostas:
7.  0 16.662 [*ponte_do_brooklyn | *cruza | *rio_east] - [14.662 1.000 1.000]
8.
9.  Lista de triplas candidatas a resposta:
10. 0 16.662 [*ponte_do_brooklyn | *cruza | *rio_east] - [14.662 1.000 1.000]
11. 1 16.162 [*ponte_do_brooklyn | latminutes | 42 | *ponte_do_brooklyn | *cruza |
    *rio_east] - [14.662 0.500 1.000]
```

Quadro 24: Retorno do sistema ENSEPRO ao ser executado com a opção de retornar a resposta à pergunta.

A listagem de melhores respostas (linha 7) e de candidatas (linhas 10 e 11) tem o seguinte padrão de composição:

<posição na lista> <valor de ranque> [tripla candidata] - [M1 M2 M3]

O item M1 corresponde à soma da métrica M1 individual de cada elemento da tripla. Os itens M2 e M3 são os valores das métricas calculadas avaliando-se atributos da tripla como um todo. O valor de ranque da tripla corresponde ao somatório dos valores das três métricas.

Pode-se perceber ainda no Quadro 24 que alguns elementos da tripla candidata estão marcados com um asterisco (“*”). A função deste asterisco é destacar visualmente os

[24] A lista completa de candidatas à resposta para a pergunta tem 28.551 triplas simples e 1.567.056 pares de triplas, totalizando 1.595.607 candidatas.

elementos da tripla que casaram lexicalmente com algum dos TRs da frase. Nas próximas seções apresentam-se os detalhes relacionados ao cálculo de cada uma das métricas.

4.3.4.1 Métrica 1 – peso modularizado da classe gramatical

A Métrica 1 (M1) está diretamente relacionada aos atributos linguísticos dos TRs com os quais casaram lexicalmente os elementos da tripla candidata. Em termos gerais, o cálculo da métrica M1 de cada elemento da tripla candidata consiste primariamente no valor de um peso associado à classe gramatical de cada tipo de TR, o qual é modularizado pelo grau de semelhança lexical entre o TR e o respectivo elemento da tripla.

Para calcular a métrica M1, o sistema consulta uma lista de pesos associados às classes gramaticais dos TRs dependendo do seu tipo. No Quadro 25 pode-se observar os pesos definidos para as classes gramaticais dos três tipos de TRs: originais da frase (linhas 3 a 5), sinônimos (linhas 8 a 10) e nominalizações (linha 7).

```
1.  "pesos": {
2.      "classes": {
3.          "substantivo_proprio": 10,
4.          "verbo": 3,
5.          "substantivo_comum": 2,
6.          "adjetivo": 1
7.          "verbo_nominalizado": 2.8,
8.          "substantivo_comum_sinonimo": 1.8,
9.          "verbo_sinonimo": 1,
10.         "verbo_nominalizado_sinonimo": 1,
11.     }
12. }
```

Quadro 25: Lista com a definição dos pesos das classes gramaticais por tipo de TR.

Os valores dos pesos têm por objetivo permitir um ajuste mais preciso do sistema de ranqueamento. O valor inicial dos pesos foi atribuído com base na importância da classe gramatical e do tipo de TR para a localização da resposta. O peso dos substantivos próprios, devido ao fato destes serem essenciais para a localização de candidatas relacionadas ao contexto da pergunta, receberam um valor muito mais alto do que os pesos atribuídos às outras classes.

No caso dos pesos associados aos TRs verbais, estes recebem um valor maior do que o peso associado aos substantivos comuns devido a maior confiança de que a função semântica do TR verbal na frase seja muito próxima da função semântica do predicado na tripla[25]. Em outras palavras, caso encontre-se uma tripla candidata contendo um predicado que case lexicalmente com um TR verbal, tem-se uma maior confiança de que esta candidata esteja relacionada semanticamente à pergunta. Esta maior confiança é traduzida no maior valor associado ao seu peso gramatical.

[25] Importante relembrar que os TRs verbais são sempre derivados de verbos relacionais e, por isto, via de regra, serão predicados nas triplas.

É exatamente a baixa confiança na função semântica do substantivo comum em relação ao seu papel na tripla que o leva a ter o menor peso entre as três classes gramaticais que podem ser TRs. Como não se pode prever qual papel os substantivos comuns podem assumir nas triplas (ou seja, não se sabe se serão sujeito, predicado ou objeto na tripla), é razoável concluir que deve-se ter menos confiança de que o papel semântico na tripla seja coerente com o papel do TR na frase.

O menor valor associado às classes gramaticais dos TRs derivados (sinônimos e nominalizações) advém da necessidade de priorizar as triplas candidatas que contenham elementos que casem com os TRs originais em detrimento àquelas que casem com os TRs derivados.

Observa-se ainda que há um peso associado aos adjetivos (linha 6 do Quadro 25), uma classe gramatical que não pode ser um TR[26]. O motivo de associar-se pesos aos adjetivos será visto em detalhes mais adiante, na seção 4.3.4.6 Quantidade de referências a adjetivos na tripla candidata.

É importante citar que, embora os valores definidos inicialmente para o sistema de pesos tenha sido fortemente influenciado pelos aspectos semânticos das classes gramaticais em relação ao sistema de representação de informações em triplas, os valores atuais são fruto de ajustes realizados no decorrer dos experimentos realizados durante o desenvolvimento do trabalho.

Como se vê, o mecanismo de pesos da métrica M1 tem por objetivo traduzir em números uma série de hipóteses em relação às classes gramaticais dos TRs e as suas funções semânticas na frase e nas triplas da BC. Considera-se que este sistema de pesos é uma das contribuições importantes deste trabalho, uma vez que permite mapear semântica e confiança em um sistema bastante claro e objetivo de representação numérica.

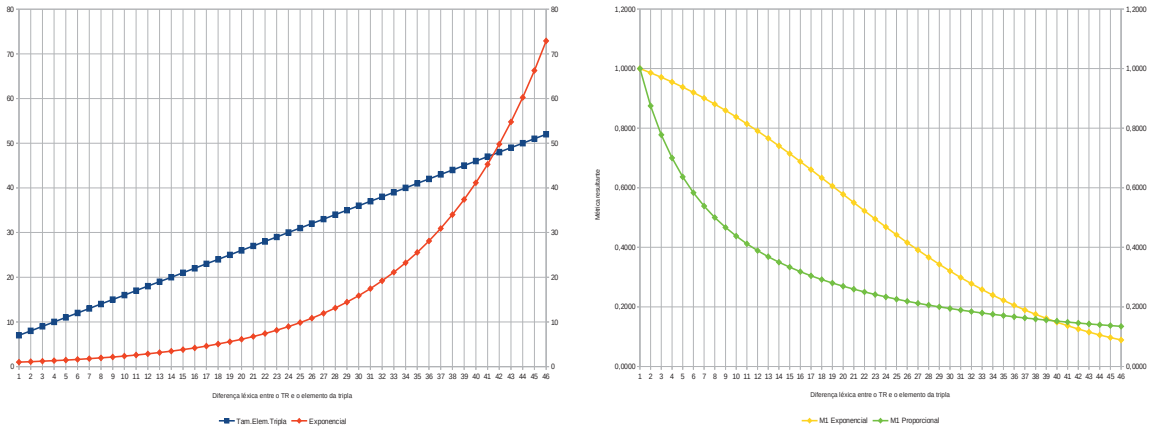
Mas o sistema de pesos não define por si só o valor da métrica M1. Pode-se observar na fórmula de cálculo da métrica M1 abaixo que o valor dos pesos das classes gramaticais é modularizado pelo grau de semelhança entre o TR e o elemento da tripla, ou seja, quanto maior a diferença léxica entre o TR e o elemento da tripla, menor será o valor da métrica M1. Caso o TR seja idêntico ao elemento da tripla, M1 será igual ao valor do peso associado ao tipo do TR.

$$M_1 = peso(TR) \times \frac{tamanho(TR)}{tamanho(TR) - 1 + 1,1^{tam(elemento\ tripla) - tam(TR)}}$$

Pode-se ver que a modularização do peso se faz pelo uso de uma função exponencial, a qual tem como objetivo suavizar ou acentuar o impacto da diferença lexical entre TR e o elemento da tripla sobre o valor final da métrica. Nos gráficos apresentados na Figura 8 pode-se observar mais claramente o efeito do uso da exponencial no cálculo da métrica M1.

[26] Somente substantivos (próprios e comuns) e verbos (relacionais) podem ser TRs.

Figura 8: Impacto do uso de exponencial no cálculo da métrica M1.



(1) (2)

Fonte: elaborada pelo autor.

No gráfico 1 da Figura 8 está representada em azul uma sequência de valores que mostra uma diferença linear crescente entre os tamanhos do TR e do elemento da tripla. Em vermelho está representado os valores que a função exponencial assume para cada uma destas diferenças. Como pode-se observar, a tendência do valor da função exponencial até 41 carácteres é ser menor que o valor da diferença. Contudo, após 41 carácteres, inverte-se este comportamento, com a função exponencial retornando valores maiores do que a diferença lexical.

No gráfico 2 da Figura 8 pode-se observar o efeito do uso da modularização do peso com uma função exponencial sobre os valores da métrica M1.(em amarelo) comparada ao uso de uma modularização por simples proporção entre a quantidade de carácteres do TR e do respectivo elemento da tripla (em verde).

Observa-se que o declínio dos valores da métrica M1 com função exponencial é mais lento até 40 carácteres de diferença. Após isto, os valores de M1 com função exponencial caem de forma mais acentuada do que a simples proporção.

O uso da função exponencial no cálculo resultou exatamente no comportamento pretendido para a métrica M1: atenuar até um certo ponto o efeito da diferença léxica sobre o peso gramatical, posteriormente acentuando este efeito.

Tabela 2: Cálculo da métrica M1 para a tripla [ponte_do_brooklyn, cruza, rio_east] em relação aos TRs da pergunta "Que rio a Ponte do Brooklyn cruza?".

Elemento da tripla	**TR associado**	**Classe Gramatical do TR**	**Peso do TR**	**M1**
ponte_do_brooklyn	ponte_do_broolyn	Substantivo próprio	10	10,000
cruza	cruza	Verbo	03	03,000
rio_east	rio	Substantivo comum	02	01,662
			M1 total:	14,662

Como se pode perceber, a métrica M1 é calculada para cada elemento da tripla candidata que casou lexicalmente com algum TR da frase. A métrica M1 da tripla candidata é o somatório dos valores obtidos para cada elemento da tripla candidata. Um exemplo de cálculo da métrica M1 para a primeira tripla candidata do Quadro 24 em relação à pergunta “Que rio a Ponte do Brooklyn cruza?” é apresentado na Tabela 2.

Os valores da métrica M1 para os elementos ponte_do_brooklyn e cruza são iguais ao peso do TR, pois os elementos da tripla são lexicalmente idênticos aos TRs com os quais casaram. Já para o elemento rio_east, que é lexicalmente diferente do TR associado, o valor de M1 é ligeiramente menor que o peso. A métrica M1 total da tripla candidata é a soma dos valores M1 atribuídos aos elementos da tripla.

4.3.4.2 Resolução de conflitos semânticos na Métrica M1

A base para o cálculo da métrica M1 é o casamento lexical entre os elementos das triplas candidatas e os TRs da pergunta. Podem haver situações em que um TR case lexicalmente com mais de um elemento da tripla candidata, gerando assim uma situação de conflito semântico, pois um TR não pode cumprir com dois papéis diferentes na mesma tripla. É necessário resolver estes conflitos semânticos antes de realizar o cálculo da métrica.

O Quadro 26 apresenta uma pergunta cujas triplas candidatas contém conflito semântico.

***Quadro 26: Exemplo de pergunta contendo conflito semântico nas triplas candidatas*.**

```
# ensepro -resposta -frase "Liste todos os musicais com músicas de Elton John."
--> Termos Relevantes:
----> TR 0: [H:n|9| músicas]
----> TR 1: [DP:prop|12| Elton_John]
Lista de triplas candidatas a resposta:
0 14.000 [*aida_(musical) | *musica | *elton_john] - [12.000 1.000 1.000]
1 14.000 [*o_rei_leao_(musical) | *musica | *elton_john] - [12.000 1.000 1.000]
2 14.000 [*billy_elliot___the_musical | *musica | *elton_john] - [12.000 1.000 1.000]
3 13.860 [*elton_john | instrumento | *teclado_(instrumento_musical)] - [11.860 1.000
1.000]
```

O conflito semântico ocorre em relação ao TR 0 nas triplas candidatas 0, 1 e 2. Como o cálculo da métrica M1 leva em consideração o casamento lexical entre o TR e os elementos das triplas candidatas, verifica-se que tanto o sujeito quando o predicado das candidatas 0, 1 e 2 casam com o TR 0.

Não é razoável esta situação, pois um TR não pode cumprir com dois papeis diferentes em uma mesma tripla. Concluí-se então que está ocorrendo um conflito semântico nesta candidata. Para calcular-se a métrica M1 das triplas candidatas 0, 1 e 2 do Quadro 26 é necessário primeiramente resolver o conflito semântico. Foram implementadas no sistema quatro políticas para resolução de conflitos semânticos: BEST_MATCH, WORST_MATCH, AVG and SUM.

Todas as políticas de resolução fundamentam-se no cálculo da métrica M1 para ambos os elementos conflitantes da tripla candidata, para então aplicar-se a política de resolução. Se a política de resolução configurada for a BEST_MATCH, então o valor da métrica M1 adotado para a tripla será a métrica M1 do elemento que melhor case lexicalmente com o TR.

A política WORST_MATCH implementa o inverso, adotando para a tripla o valor da métrica M1 do elemento com menor casamento lexical. As políticas AVG e SUM adotam respectivamente a média e a soma das métricas M1 para os elementos conflitantes. Pode-se perceber na Tabela 3 que a política de resolução BEST_MATCH foi utilizada para resolver os conflitos semânticos do exemplo mostrado no Quadro 26, uma vez que o valor da métrica M1 do elemento "aida_(musical)" foi ignorado no cálculo da métrica M1 da tripla.

Tabela 3: Cálculo da métrica M1 para a tripla "[aida_(musical), musica, elton_john]" em relação aos TRs da frase "Liste todos os musicais com músicas de Elton John." utilizando a política de resolução de conflitos semânticos BEST_MATCH.

Elemento da tripla	TR associado	Classe Gramatical do TR	Peso do TR	M1
aida_(musical)	musica	Substantivo comum	02	01,761
musica	musica	Substantivo comum	02	02,000
elton_john	elton_john	Substantivo próprio	10	10,000
			M1 total:	12,000

Do ponto de vista semântico, a escolha da política BEST_MATCH para a resolução do conflito fez com que o TR 0 (musica) fosse associado somente ao predicado da tripla candidata 0, fazendo com que o sistema ignorasse no momento de calcular a métrica M1 o casamento lexical do TR 0 com o sujeito da tripla.

A política de resolução de conflitos impacta direta e profundamente nos cálculos da métrica M1, do ranque das triplas e, por consequência, também na escolha da resposta correta. O impacto da política de resolução de conflitos semânticos será visto na seção 4.3.5 Seleção de respostas.

As quatro políticas de resolução do conflito semântico foram implementadas no ENSEPRO e são ativadas com base em configuração, o que possibilita a fácil adaptação do sistema de resolução de conflito semântico a diferentes contextos ou domínios.

4.3.4.3 Cálculo da métrica M1 para candidatas compostas

Os procedimento para o cálculo da métrica M1 descritos até agora aplicam-se à triplas candidatas simples. Existem algumas poucas regras adicionais ao calcular-se a métrica M1 quando a candidata é composta por mais de uma tripla.

Conforme comentado anteriormente, decidiu-se que atualmente são geradas somente pares de triplas candidatas, a fim de manter-se o tempo de resposta do ENSEPRO frente a BC extensas dentro de um padrão aceitável para sistemas de PR.

Contudo, para BCs não tão extensas, poderia-se gerar candidatas com maior quantidade de triplas, como trincas ou quadras, por exemplo. É importante então destacar que

as regras de cálculo da métrica M1 explicadas nesta seção são genéricas e aplicam-se a qualquer número de triplas candidatas, não somente a pares.

Triplas candidatas a resposta compostas por mais de uma tripla são relacionadas pelo sujeito ou objeto, conforme visto na seção 4.3.3 Geração combinatorial de candidatas à resposta. No exemplo apresentado no Quadro 27 pode-se observar na lista de triplas que a candidata 1 compõe-se de um par de triplas relacionadas pelo mesmo sujeito: "[ponte_do_brooklyn | latminutes | 42]" e "[ponte_do_brooklyn | cruza | rio_east]".

Quadro 27: Cálculo da métrica M_1 para candidatas à resposta contendo triplas simples e pares de triplas.

```
# ensepro  -tr -original -resposta -frase "Que rio a Ponte do Brooklyn cruza?"
--> Termos Relevantes:
----> TR 0: [H:n|4| rio]
----> TR 1: [H:prop|7| Ponte_do_Brooklyn]
----> TR 2: [P:v-fin|8| cruza]
Melhores respostas:
0 16.662 [*ponte_do_brooklyn | *cruza | *rio_east] - [14.662 1.000 1.000]

Lista de triplas candidatas a resposta:
0 16.662 [*ponte_do_brooklyn | *cruza | *rio_east] - [14.662 1.000 1.000]
1 16.162 [*ponte_do_brooklyn | latminutes | 42 | *ponte_do_brooklyn | *cruza | *rio_east] -
[14.662 0.500 1.000]
```

Por conta disto, candidatas a resposta com mais de uma tripla sempre vão conter ou o sujeito ou o objeto repetidos, sendo possível inclusive haver a repetição de ambos simultaneamente. Esta repetição deve ser levada em conta no momento de calcular-se a métrica M1 da tripla composta.

Pode-se observar que a tripla 1 do Quadro 27, apesar de conter duas referências ao TR "ponte_do_brooklyn", tem a métrica M1 igual a da tripla 0. Verifica-se então que o algoritmo para cálculo da métrica M1 leva em consideração somente se houve ou não referência ao TR, não a quantidade de vezes que o TR é referido dentro da tripla.

Verifica-se que esta forma de calcular a métrica M1 corrobora significativamente na localização da resposta correta, como no exemplo dado no Quadro 27. Caso as referências duplicadas fossem levadas em consideração no momento do cálculo da métrica M1, a tripla simples contendo a resposta correta (candidata 0) teria o seu valor de ranqueamento menor que as triplas compostas (como a candidata 1), o que impediria o sistema de localizar a resposta correta.

A métrica M1 é a métrica com procedimento mais complexo. As outras duas métricas tem um procedimento bastante mais simples, como pode ser visto nas duas próximas seções.

4.3.4.4 Métrica 2 – Proporção de referências a TRs na tripla candidata

A Métrica 2 (M2) é uma simples proporção de referências únicas da tripla aos TRs da pergunta em relação ao número de elementos da candidata. A métrica M2 é o segundo valor a compor o ranque da tripla candidata.

Na lista de triplas candidatas do Quadro 27, por exemplo, a métrica M2 para a candidata 0 é 1 (3 / 3), enquanto para a candidata 1 o valor de M2 é 0,5 (3 / 6). Percebe-se que novamente as referências repetidas são ignoradas no momento de realizar o cálculo da métrica.

Um exemplo mais interessante para se analisar a importância da métrica M2 na localização da resposta correta é apresentado no Quadro 28. Percebe-se neste exemplo que o sistema de derivação de TRs não foi suficiente para que o sistema conseguisse relacionar todos os elementos da tripla aos TRs da frase.

Pode-se ver que o predicado "elenco" não foi associado ao TR verbal "atuar". Isto ocorreu porque o sistema de derivação implementado no sistema ENSEPRO não consegue estabelecer uma relação léxica e nem semântica entre o TR e o predicado da tripla, pois não há na lista de derivações (nem nos sinônimos e nem nas nominalizações) a palavra elenco.

Observa-se no exemplo apresentado no Quadro 28 que é a métrica M2 (0,667) que possibilita ao sistema localizar a resposta correta para a pergunta, pois os valores de M1 e M2 (20 e 1 respectivamente) são idênticos para as candidatas 0 e 1.
Assim como a métrica M2, a última métrica a compor o valor do ranque da candidata também é uma proporção, mas desta vez relacionada a classe gramatical dos TRs.

Quadro 28: Cálculo da métrica M_1 para candidatas à resposta contendo triplas simples e pares de triplas.

```
# ensepro -tr -final -sin -nom -frase "Christian Bale atua em Batman Begins?"
--> Termos Relevantes:
----> TR 0: [S:prop|2| Christian_Bale]
--------> Sinonimos: {'por': [], 'eng': []}
----> TR 1: [P:v-fin|3| atua]
--------> Sinonimos: {'por': [parecer, fazer_papel, representar, fazer, agir, comportar-se,
atuar], 'eng': [behave, act, do, roleplay, playact, play]}
--------> Nominalizacoes: [ator | atuação | atuador]
----> TR 2: [DP:prop|6| Batman_Begins]
--------> Sinonimos: {'por': [], 'eng': []}Melhores respostas:
0 21.667 [*batman_begins | elenco | *christian_bale] - [20.000 0.667 1.000]

Lista de triplas candidatas a resposta:
0 21.667 [*batman_begins | elenco | *christian_bale] - [20.000 0.667 1.000]
1 21.333 [*christian_bale | goldenglobe | *melhor ator coadjuvante em cinema |
*batman_begins | elenco | *christian_bale] - [20.000 0.333 1.000]
```

4.3.4.5 Métrica 3 – Proporção de substantivos próprios na tripla candidata

Como já citado anteriormente em diversos pontos deste capítulo, os substantivos próprios da pergunta são imprescindíveis para a localização da resposta correta. Todos os substantivos próprios referenciados na frase sempre obrigatoriamente aparecem na tripla que responde corretamente a pergunta.

Triplas candidatas que não referenciem todos os substantivos próprios não podem conter uma resposta correta. A métrica M3 tem como principal objetivo fazer com que as triplas que não contenham todos substantivos próprios sejam levadas para baixo na lista ordenada pelo ranque. A fórmula para o cálculo da métrica M3 é apresentada abaixo.

$$M_3 = \frac{Quantidade\ de\ susbst\ próprios\ da\ tripla\ candidata}{Quantidade\ de\ susbst\ próprios\ da\ frase}$$

Embora o valor de M3 seja o último componente considerado como métrica da tripla, o sistema de ranqueamento ainda leva em conta a presença de referências léxicas aos adjetivos da pergunta. Na próxima seção explica-se porque este último componente não é considerado como uma métrica da tripla.

4.3.4.6 Quantidade de referências a adjetivos na tripla candidata

Uma situação bastante curiosa em relação ao sistema de métricas criado para ranquear as triplas candidatas a resposta ocorreu em relação ao último critério. Desde que foi elaborado, o sistema de métrica teve poucas alterações no decorrer dos experimentos e manteve-se praticamente imutável desde a sua concepção inicial. Verificou-se no entanto uma situação bastante interessante no decorrer dos experimentos em relação a perguntas contendo adjetivos.

Como já citado anteriormente, adjetivos não são considerados como TR da frase, uma vez que somente substantivos e verbos são relevantes para a busca de triplas candidatas à resposta na BC. Verificou-se, no entanto, que triplas contendo referências a adjetivos deveriam ganhar destaque na lista de candidatas, pois comumente continham a resposta correta.

Diante desta constatação empírica, alterou-se o algoritmo de seleção de termos relevantes de forma a incluir os adjetivos na lista de TRs. O resultado obtido foi bastante negativo, resultando em listas de candidatas com triplas que, de forma alguma, poderiam contribuir para a localização da resposta.

Verificou-se na prática que a inclusão dos adjetivos na lista de TRs resultou em uma lista de candidata com uma grande quantidade de triplas que não tinham relação com a pergunta. Em um estudo mais aprofundado da questão, constatou-se que nem todos os adjetivos poderiam ser considerados como TR, mas somente um tipo específico de adjetivo: aqueles adjetivos que, dependendo do contexto, podem ser usados como um substantivo comum.

É o caso da palavra “musical” na frase apresentada no Quadro 29. Esta palavra pode, dependendo do contexto, ser classificada como um adjetivo ou um substantivo. O parser utilizado para anotar linguisticamente as frases diferencia este grupo de adjetivos, permitindo então a implementação de um novo algoritmo, o qual somente considerava como TR os adjetivos passíveis desta dupla função gramatical.

***Quadro 29: Exemplo de pergunta contendo adjetivos substantivados*.**

```
# ensepro -resposta -frase "Liste todos os musicais com músicas de Elton John."
--> Termos Relevantes:
----> TR 0: [H:n|9| músicas]
----> TR 1: [DP:prop|12| Elton_John]
Lista de triplas candidatas a resposta:
0 15.000 [*aida_(musical) | *musica | *elton_john] - [13.000 1.000 1.000]
1 15.000 [*o_rei_leao_(musical) | *musica | *elton_john] - [13.000 1.000 1.000]
2 15.000 [*billy_elliot___the_musical | *musica | *elton_john] - [13.000 1.000 1.000]
3 13.860 [*elton_john | instrumento | *teclado_(instrumento_musical)] - [11.860 1.000 1.000]
```

O resultado deste novo algoritmo de seleção de TRs apresentou um desempenho tão ruim quanto o anterior, produzindo listas de candidatas a resposta de baixa qualidade. Diante do resultado ruim, desistiu-se da inclusão de adjetivos na lista de TRs. Este é o motivo pelo qual a palavra “musicais” não aparece como sendo um TR mesmo sendo um substantivo da frase apresentada no Quadro 29.

Embora resolvida a questão da seleção dos TRs, causou uma certa inquietação o fato de haver surgido um substantivo comum (pois esta é a classe gramatical da palavra “musical” no contexto da frase apresentada no Quadro 29) que não deveria ser incluído na lista de TRs.

Diante da situação, resolveu-se fazer mais uma experiência: se os adjetivos substantivados fossem ignorados no momento de selecionar as triplas candidatas mas fossem considerados no momento do ranqueamento? O resultado do experimento foi muito positivo, fazendo com que triplas candidatas contendo a resposta correta fossem significativamente destacadas na lista de candidatas.

Como mostra o Quadro 29, a métrica M1 para as três primeiras triplas da lista de candidatas somam agora 13. No detalhamento do cálculo da métrica M1 para a candidata 0 apresentado na Tabela 4 pode-se ver que agora o peso associado ao adjetivo está sendo considerado.

Tabela 4: Cálculo da métrica M1 para a tripla "[aida_(musical) | musica | elton_john]" em relação aos TRs da frase “Liste todos os musicais com músicas de Elton John.” contabilizando-se o peso do adjetivo.

Elemento da tripla	**Palavra da frase**	**Classe Gramatical**	**Peso**	**M1**
aida_(musical)	musical	Adjetivo	01	01,000
musica	musica	Substantivo comum	02	02,000
elton_john	elton_john	Substantivo próprio	10	10,000
			M1 total:	13,000

É importante notar que diferentemente do cálculo realizado para os pesos dos TRs, o peso do adjetivo é sempre contabilizado na sua totalidade, ou seja, não se realiza a modularização do peso de acordo com a semelhança lexical do adjetivo e o elemento da tripla. No caso do exemplo apresentado na Tabela 4, mesmo sendo o adjetivo “musical”

lexicalmente diferente do elemento “aida_(musical)”, a métrica M1 recebeu o peso integral da classe gramatical.

Geralmente, a diferença de valor de ranque entre as respostas corretas na lista de candidatas é bastante pequena, usualmente com diferenças fracionárias menores que um. Ao utilizar os adjetivos (indistintamente), verificou-se uma melhora significativa no ranqueamento.

A verificação das referências a adjetivos nas candidatas é o último componente do sistema de ranqueamento para as triplas candidatas. O próximo processo a ser executado pelo módulo CBC é a seleção das respostas que serão retornadas pelo sistema ENSEPRO, assunto da próxima seção.

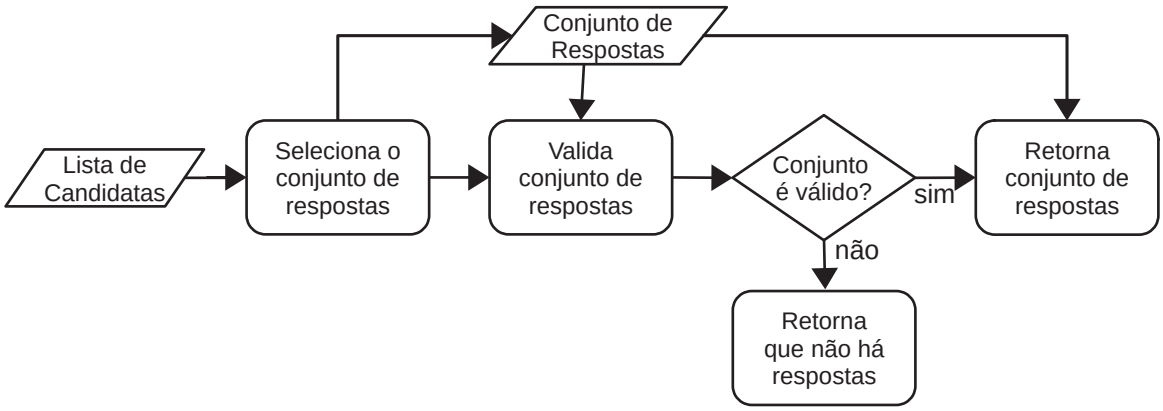
4.3.5 Seleção de respostas

Após o cálculo do ranque de cada uma das triplas, a lista de candidatas é colocada em ordem decrescente de ranque. O objetivo de ranquear e ordenar a lista de candidatas é colocar no início da lista as candidatas que, segundo os critérios de ranqueamento, contém a “melhor resposta” encontrada pelo sistema na BC para a questão.

Chama-se de “melhor resposta” porque não necessariamente a tripla no início da lista é a resposta correta para a questão recebida. Pode ocorrer que a BC não contenha a resposta para a questão ou que o sistema de busca e ranqueamento não tenha obtido sucesso na localização da resposta. Caso uma destas duas situações ocorra, a tripla no início da lista de candidatas não seria a resposta correta para a questão, mas apenas o melhor que o sistema pode encontrar.

O esquema geral do fluxo de processos de seleção e validação das melhores respostas está representado na Figura 9. O primeiro passo para determinar se a resposta foi encontrada é identificar o conjunto de triplas candidatas que melhor respondem a questão.

Figura 9: Processo de seleção e validação do conjunto de respostas.



Fonte: elaborada pelo autor.

Resumidamente, o conjunto de melhores respostas é composto preliminarmente pelas triplas que obtiveram o maior ranque. Posteriormente, são adicionadas as candidatas da lista que seguirem o padrão lexical das triplas que já estão no conjunto de melhores respostas.

Submete-se então o conjunto de melhores respostas ao processo de validação, o qual tem por objetivo avaliar se o conjunto de candidatas selecionado é aceitável como resposta para a questão. O processo de validação avalia individualmente a admissibilidade das triplas do conjunto. O final do processo de validação é o retorno da resposta à pergunta, caso uma tenha sido encontrada.

O detalhamento dos processos de seleção e validação do conjunto de respostas são vistos em detalhes nas próximas seções.

4.3.5.1 Seleção do conjunto de melhores respostas

O processo de seleção da resposta inicia pela busca do conjunto de melhores respostas na lista de candidatas. Preliminarmente, as melhores respostas para a questão são aquelas candidatas que obtiveram o maior valor de ranqueamento.

No Quadro 30 são apresentados dois exemplos de questões e suas respectivas listas de candidatas a resposta, destacando-se em negrito os respectivos conjuntos de melhores respostas. Para o exemplo 1, o conjunto preliminar de melhores respostas compõe-se de somente uma candidata, uma vez que o seu ranque (16,662) é o maior dentre as candidatas. No caso do exemplo 2, no entanto, o conjunto preliminar é formado por duas candidatas, uma vez que ambas obtiveram o maior valor de ranque (13.667) dentre as candidatas.

***Quadro 30: Exemplo de pergunta e respectivas listas de candidatas a resposta, destacando-se em negrito o conjunto inicial de melhores respostas*.**

Exemplo 1
ensepro -resposta -frase "Que rio a Ponte do Brooklyn cruza?"
Lista de triplas candidatas a resposta:
0 16.662 [*ponte_do_brooklyn | *cruza | *rio_east] - [*14.662* 1.000 1.000]
1 16.162 [*ponte_do_brooklyn | latminutes | 42 | *ponte_do_brooklyn | *cruza | *rio_east] - [*14.662* 0.500 1.000]

Exemplo 2:
ensepro -frase "Quem criou Batman" -resposta
Lista de triplas candidatas a resposta:
0 13.667 [*batman | *criador | bob_kane] - [12.000 0.667 1.000]
1 13.667 [*batman | *criador | bill_finger] - [12.000 0.667 1.000]
2 13.503 [dick_grayson | *criador | *batman:] - [11.836 0.667 1.000]

Obtido o conjunto preliminar de melhores respostas, inicia-se o processo de seu refinamento. O processo de refinamento do conjunto preliminar é realizado com base na análise do que convencionou-se chamar de padrão de semelhança lexical das candidatas. Na Tabela 5 podem ser vistos os padrões de semelhanças lexicais gerados a partir dos exemplos apresentados acima.

O padrão de semelhança lexical compõe-se da sequência de carácteres em comum entre o elemento da tripla e o respectivo TR, seguida por uma letra que indica o tipo de casamento léxico ocorrido. Se o elemento da tripla candidata for idêntico ao TR, então o seu tipo do casamento léxico é total, sendo representado pela letra “t”. Caso o elemento da tripla case parcialmente com o TR, então o seu tipo será representado pela letra “p”.

Tabela 5: Exemplos de triplas candidatas e respectivos padrões de semelhança lexical.

Ex.	TRs	Tripla do conjunto preliminar	Padrões de semelhança lexical
1	ponte_do_brooklyn, cruza, rio	[ponte_do_brooklyn, cruza, rio_east]	ponte_do_brooklyn-t cruza-t rio-p
2	batman, criador	[batman, criador, bob_kane]	batman-t criador-t
2	batman, criador	[batman, criador, bill_finger]	batman-t criador-t

Nos exemplos apresentados na Tabela 5, houve casamento léxico total para os TRs “ponte_do_brooklyn” e “cruza” no exemplo 1 (padrões de semelhança lexical “ponte_do_brooklyn-t” e “cruza-t”) e para os TRs “batman” e “criador” no exemplo 2 (padrões de semelhança lexical “batman-t” e “criador-t”). No entanto, no caso do TR “rio” do exemplo 1 houve um casamento léxico parcial com o elemento “rio_east” (representado pelo padrão de semelhança lexical “rio-p”).

Depois de gerados os padrões de semelhança lexical, o próximo passo é identificar o melhor padrão dentre as candidatas do conjunto preliminar. Para identificar-se o melhor padrão usa-se como critério de avaliação um escore gerado a partir dos tipos de casamento lexical que ele possui.

A cada casamento léxico total encontrado no padrão soma-se ao seu escore o valor 1,1 e a cada casamento parcial soma-se 0,5. Na Tabela 6 pode-se ver como são realizados os cálculos do escore para os padrões de semelhança lexical dos exemplos 1 e 2 apresentados acima.

Tabela 6: Demonstração do cálculo do escore do padrão de semelhança lexical.

Ex.	Tripla do conjunto preliminar	Padrão semelhança lexical	Escore do padrão
1	[ponte_do_brooklyn, cruza, rio_east]	ponte_do_brooklyn-t cruza-t rio-p	1,1 + 1,1 + 0,5 = 2,7
2	[batman, criador, bob_kane]	batman-t criador-t	1,1 + 1,1 = 2,2
2	[batman, criador, bill_finger]	batman-t criador-t	1,1 + 1,1 = 2,2

Feitos os cálculos, escolhe-se o melhor padrão de semelhança lexical pelo maior escore. Procede-se então com o refinamento do conjunto preliminar de melhores respostas através da remoção de todas as candidatas cujo o padrão de semelhança lexical seja diferente do melhor padrão.

Analisando-se o algoritmo acima, percebe-se que o conjunto preliminar de melhores respostas restringe-se àquelas candidatas com melhor ranque. Contudo, podem ocorrer casos em que a resposta à questão está fora do conjunto preliminar de melhores respostas, como no exemplo apresentado no Quadro 31.

Quadro 31: Exemplo de pergunta em que as melhores respostas tem ranques diferentes.

ensepro -resposta -frase "Liste todos os musicais com músicas de Elton John."
--> Termos Relevantes:
----> TR 0: [H:n|9| músicas]
----> TR 1: [DP:prop|12| Elton_John]
Lista de triplas candidatas a resposta:
0 16.680 [*aida_(musical) | *musica | *elton_john] - [14.680 1.000 1.000]
1 16.364 [*o_rei_leao_(musical) | *musica | *elton_john] - [14.364 1.000 1.000]
2 16.023 [*billy_elliot___the_musical | *musica | *elton_john] - [14.023 1.000 1.000]
3 13.860 [*elton_john | instrumento | *teclado_(instrumento_musical)] - [11.860 1.000 1.000]

No exemplo acima, o conjunto preliminar de melhores respostas seria composto somente pela tripla 0, uma vez que esta é a única candidata com o maior ranque da lista (16,680). No entanto, são respostas também para a questão as triplas 1 e 2, as quais, devido ao seu valor de ranque serem menores do que a candidata 0, não são incluídas no conjunto preliminar de melhores respostas.

É por isto que, após o processo de refinamento do conjunto preliminar, faz-se uma busca nas candidatas subsequentes da lista para identificar outras candidatas cujo padrão de semelhança lexical seja igual ao melhor padrão. Estas candidatas são incluídas no conjunto definitivo de melhores respostas. Como se pode ver na Tabela 7, o padrão de semelhança lexical das triplas 1 e 2 é idêntico ao da tripla 0, motivo pelo qual elas são incluídas no conjunto definitivo de melhores respostas.

Tabela 7: Padrão de semelhança lexical para o conjunto definitivo de melhores respostas para a frase "Liste todos os musicais com músicas de Elton John."

Tripla	Triplas do conjunto definitivo	Padrão semelhança lexical	Escore
0	[aida_(musical) , musica, elton_john]	musical-p musica-t elton_john-t	0,5 + 1,1 + 1,1 = 2,7
1	[o_rei_leao_(musical) , musica, elton_john]	musical-p musica-t elton_john-t	0,5 + 1,1 + 1,1 = 2,7
2	[billy_elliot___the_musical , musica, elton_john]	musical-p musica-t elton_john-t	0,5 + 1,1 + 1,1 = 2,7

Definido o conjunto de melhores respostas para a questão, o módulo CBC vai realizar o último procedimento do seu fluxo de processos: a validação das respostas.

4.3.5.2 Seleção das respostas por embedding

Verificou-se algumas situações em que o processo de expansão dos TRs não gera termos derivados que seja capaz de compensar a variação linguística inerente da LN. Resumidamente, a pergunta contém palavras que não fazem parte do vocabulário da ontologia e que o processo de expansão (sinônimos e nominalização) não é capaz de fornecer termos derivados que permitam ligar os TRs a elementos das triplas.

A partir da análise do padrão de semelhança lexical do conjunto de melhores respostas é possível identificar-se algumas situações em que o problema de variação linguística causou

a impossibilidade do sistema identificar a resposta correta. Quando, ao final do processo de seleção linguístico, o padrão de semelhança lexical do conjunto definitivo de melhores respostas contém somente um item e a lista de TRs originais da frase contém ao menos dois elementos, então pode-se inferir que o problema de variação linguística ocorreu.

A pergunta apresentada no Quadro 32 representa um caso do problema de variação linguística. Podemos identificar que o conjunto preliminar de melhores respostas conterá as 44 primeiras triplas da lista de candidatas, pois todas têm o maior valor de ranque (11,333).

Quadro 32: Exemplo de pergunta em que o problema de variação linguística demanda o uso da seleção de resposta por embedding.

```
# ensepro -resposta -frase "Quem foi casada com o presidente Chirac?"
--> Termos Relevantes:
----> TR 0: [Vm:v-pcp|5| casada]
----> TR 1: [DN:prop|11| Chirac]

Melhores respostas:
0 11.333 [*jacques_chirac | esposa | bernadette_chirac] - [10.000 0.333 1.000]

Lista de triplas candidatas a resposta:
0 11.333 [*jacques_chirac | ordempresidente | 22] - [10.000 0.333 1.000]
1 11.333 [*jacques_chirac | antes | francois_mitterrand] - [10.000 0.333 1.000]
2 11.333 [*jacques_chirac | titulo | co-principe de andorra] - [10.000 0.333 1.000]
⋮
16 11.333 [*jacques_chirac | esposa | bernadette_chirac] - [10.000 0.333 1.000]
17 11.333 [*jacques_chirac | depois | raymond_barre] - [10.000 0.333 1.000]
18 11.333 [*jacques_chirac | antestitulo | francois_mitterrand] - [10.000 0.333 1.000]
⋮
43 11.333 [francois_mitterrand | depoistitulo | *jacques_chirac] - [10.000 0.333 1.000]
44 3.147 [nicolau_nikolaevich | *casareal | casa_de_romanov] - [2.814 0.333 0.000]
45 3.147 [jorge_negro | *casareal | casa_de_karaorevic] - [2.814 0.333 0.000]
⋮
```

Pode-se ver na Tabela 8 que o padrão de semelhança lexical do conjunto de melhores respostas contém somente um item, indicando que todas as triplas do conjunto têm casamento lexical com somente um dos TRs da frase.

Tabela 8: Padrão de semelhança lexical para o conjunto de melhores respostas que evidencia o problema de variação linguística.

Tripla	**Triplas do conjunto definitivo**	**Padrão lexical**	**Similaridade com “casada”**
0	[*jacques_chirac | ordempresidente | 22]	jacques_chirac-t	0.486667
⋮	⋮	⋮	⋮
16	[*jacques_chirac | esposa | bernadette_chirac]	jacques_chirac-t	0.8264854

⋮	⋮	⋮	⋮
43	[francois_mitterrand | depoistitulo | *jacques_chirac]	jacques_chirac-t	0.61557704

O fato do conjunto de melhores respostas conter triplas que tem somente um casamento lexical somado à constatação de haver mais de um TR original na frase sinalizam indubitavelmente que o problema de variação ocorreu e o método de seleção por características linguísticas não obteve sucesso em encontrar a resposta para a questão.

Constatado o problema de variação linguística, pode-se concluir que o conjunto de triplas não contém as melhores respostas, mas sim as melhores candidatas a resposta. É necessário selecionar dentro do conjunto as triplas que respondem a questão. O módulo CBC aciona então o método de seleção por embedding, que consiste na avaliação da similaridade entre TRs originais “livres” e os elementos “livres” da tripla (ou seja, aqueles que ainda não casaram lexicalmente).

Na última coluna da Tabela 8 pode-se observar que a similaridade calculada entre o único TR livre da frase (“casada”) e alguns dos elementos “livres” das triplas do conjunto de melhores respostas. Pode-se ver no Quadro 32 que o sistema de seleção de respostas por embedding seleciona a tripla 16 como melhor resposta do conjunto, pois ela apresenta a maior similaridade com o TR livre da frase.

Uma vez selecionada a resposta a ser retornada pelo sistema, o módulo CBC aciona o último processo do sistema: o processo de validação da resposta selecionada.

4.3.5.3 Validação das respostas

O processo de validação do conjunto de melhores respostas consiste na avaliação do quão semanticamente relacionados estão os TRs originais da frase e os elementos das triplas que compõem o conjunto definitivo de melhores respostas.

A validação das respostas é necessária porque o fundamento do algoritmo de seleção de respostas é fortemente baseado no nível léxico dos TRs (originais e derivados) da frase, influenciado pelo nível sintático no momento da consulta à BC. Como se sabe, somente estes dois níveis de processamento da LN não são suficientes para captar-se o significado das palavras.

Ou seja, o conjunto definitivo de melhores respostas pode conter triplas que tenham somente uma relação léxica com a pergunta, até mesmo sintática, mas não semântica. Ou seja, o algoritmo de seleção encontrou uma tripla que contém palavras utilizadas na pergunta, mas que não tem relação semântica com o assunto da questão e, portanto, não são as respostas para a questão.

O módulo CBC utiliza word embedding para avaliar a relação semântica entre a pergunta recebida e o conjunto definitivo de melhores respostas. A validação consiste na avaliação da similaridade dos TRs originais da frase em relação a cada uma das triplas que compõem o conjunto definitivo de melhores respostas.

Na Tabela 9 pode-se observar os níveis de similaridade para cada uma das triplas do conjunto definitivo de melhores respostas para a questão do Quadro 31. Considera-se como

resposta válida àquelas triplas que obtiverem um nível de similaridade mínimo. Atualmente, este nível mínimo de similaridade está definido em 0,8.

Tabela 9: Similaridade semântica entre as triplas do conjunto definitivo de melhores respostas e os TRs originais da frase "Liste todos os musicais com músicas de Elton John.".

Tripla	Triplas do conjunto definitivo	TRs originais da frase	Similaridade
0	aida, musical, música, elton, john	música elton john	0,9185489
1	o, rei, leão, musical, música, elton, john	música elton john	0,8283860
2	billy, elliot, the, musical , música, elton, john	música elton john	0.9337313

Finaliza-se então o processo com o sistema ENSEPRO retornando todas as triplas validadas do conjunto definitivo como resposta para a questão. No próximo capítulo apresenta-se o resultado obtido em alguns experimentos realizados para avaliar o sistema.

5 IMPLEMENTAÇÃO

Com o objetivo de avaliar o modelo de sistema de PRS proposto nesta tese, implementou-se um protótipo da aplicação, o qual recebeu o nome de ENSEPRO. Este capítulo apresenta os detalhes técnicos do sistema ENSEPRO, uma visão geral da sua arquitetura e seus principais componentes.

O protótipo do sistema foi implementado como uma aplicação executada em linha de comando, a qual recebe alguns parâmetros que definem o nível de detalhamento das informações retornadas pelo ENSEPRO no decorrer dos processos realizados para a localização da resposta. No Quadro 33 é mostrada a execução do sistema com a opção de listar as opções disponíveis.

```
# ensepro -h
Usage: ensepro [-h] [-frase FRASE] [-arquivo-frases ARQUIVO_FRASES]
               [-save-json] [-save-txt] [-tr] [-sin]
               [-cn] [-lv] [-arvore] [-tags] [-resposta] [-verbose]
               [-quiet] [-original] [-final] [--sem-resposta]
               [-somente-resposta]
  -h, --help            show this help message and exit
  -frase FRASE          Frase a ser analisada. (default: None)
  -arquivo-frases ARQUIVO_FRASES
                        Arquivo contento frases a serem analisadas. (default:
                        None)
...
```

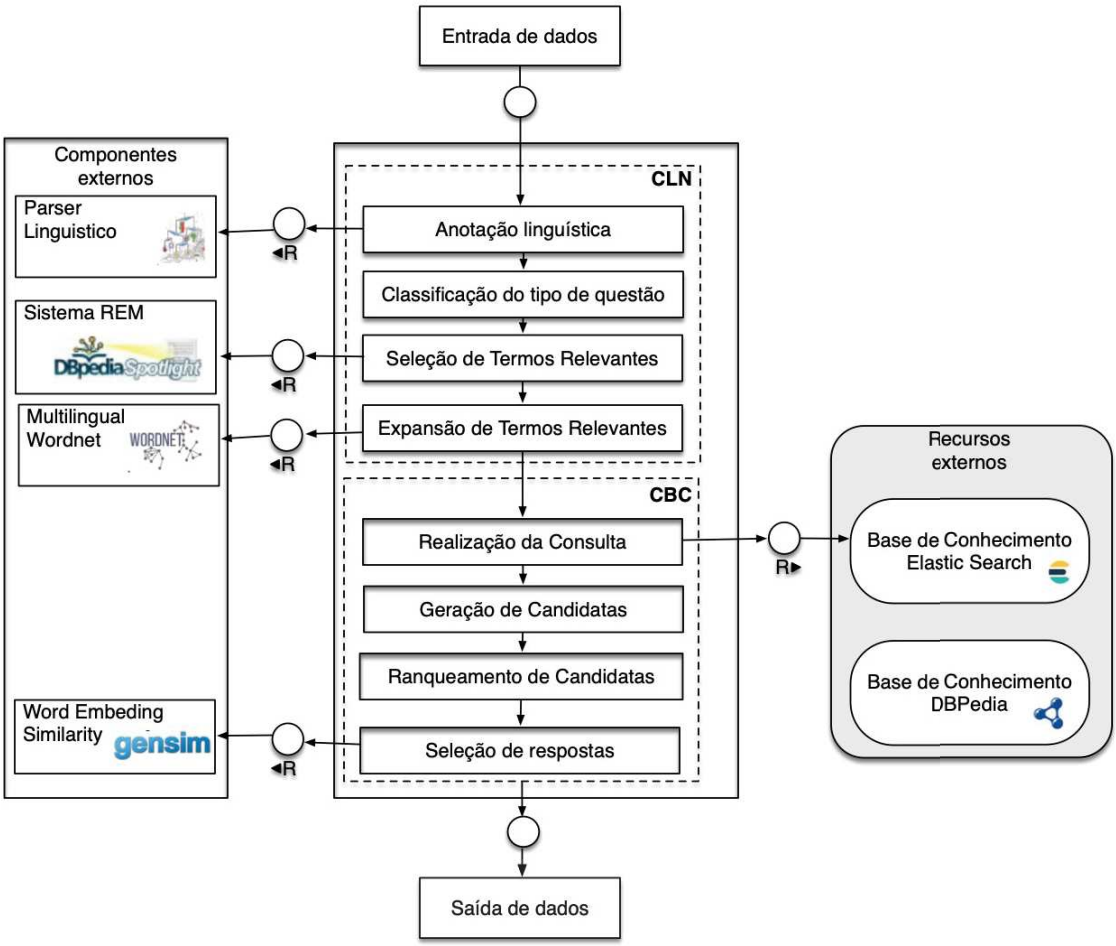
Quadro 33: lista de opções disponíveis para execução do sistema ENSEPRO.

No Quadro 33 observa-se primeiramente uma versão resumida com todas as opções disponíveis e em seguida seria exibida uma versão longa, com um texto descritivo mais detalhado sobre cada uma das opções. Por ser bastante longo o texto com a versão estendida, somente as três primeiras descrições estão sendo apresentadas aqui. No Anexo A apresenta-se a versão completa das opções do sistema ENSEPRO.

Conforme visto em vários exemplos apresentados no decorrer do capítulo 4, a chamada ao sistema ENSEPRO ocorre com o disparo do comando em linha, passando-se a frase ou arquivo de frases de perguntas a serem processadas. Este protótipo do sistema foi implementado atualmente como uma aplicação em linha de comando pois pretende-se que o ENSEPRO seja utilizado como um serviço a ser utilizado por outros sistemas.

A Figura 10 descreve a arquitetura geral utilizada para conectar os diferentes módulos desenvolvidos. A notação TAM (Technical Architecture Modeling) definida pela SAP com o propósito de cobrir tanto o nível conceitual como o nível de design de aplicações de software foi utilizada na sua descrição (KNÖPFEL, 2007).

Figura 10: Visão geral da arquitetura do ENSEPRO.



Fonte: elaborada pelo autor.

O sistema ENSEPRO compõe-se de dois módulos principais, CLN e CBC, desenvolvido em sua maior parte na linguagem de programação Python, com dois processos implementados na linguagem Java por questão de desempenho. No decorrer do fluxo de processos são utilizados cinco componentes externos ao sistema, acionados via chamada de API. Estes componentes externos são serviços chamados pelos processos que compõem o ENSEPRO: (1) o parser linguístico Palavras, (2) o sistema REM DBPedia Spotlight, (3) a ontologia Multilingual Wordnet, (4) o cache de otimização de consulta à BC em Elasticsearch e (5) o serviço de similaridade baseado em word embedding.

A comunicação com o parser linguístico tem como objetivo realizar a anotação linguística da pergunta. Esta API de comunicação foi implementada em Python e tem como principal função abstrair o parser linguístico utilizado para realizar as anotações, padronizando a comunicação e possibilitando a escolha do parser a ser utilizado. O protótipo implementado atualmente utiliza como parser linguístico o Palavras (BICK, 2000).

O ENSEPRO utiliza o algoritmo de REM do sistema DBPedia Spotlight no processo de seleção dos termos relevantes da pergunta. Optou-se pelo uso de uma versão Docker do

Spotlight[27], a qual disponibiliza localmente uma API de consulta. O processo de seleção dos termos submete a pergunta ao algoritmo de REM desta versão local do Spotlight. Foram configurados dois contêineres contendo as versões em português e em inglês do Spotlight, já que o ENSEPRO utiliza a versão em inglês caso a versão em português não reconheça nenhuma entidade (conforme explicado na seção 4.2.2 Submissão da pergunta ao sistema de REM).

O uso dos synsets da Open Multilingual Wordnet[28] (OMW) para expansão da lista de termos relevantes ocorre pela interface de software disponibilizada pela biblioteca NLTK[29], conforme visto na seção 4.2.5 Expansão dos Termos Relevantes. A geração de sinônimos em inglês e português foi implementada no ENSEPRO via esta interface do NLTK à OMW. A necessidade de otimização das consultas por parte do sistema ENSEPRO foi atendida pelo uso do Elasticsearch[30], um servidor de busca distribuído de código aberto baseado no Apache Lucene[31]. Conforme descrito na seção 5.1.1 Pré-processamento da Base de Conhecimento, a BC semântica é previamente carregada no Elasticsearch para fins de otimização do tempo de resposta do ENSEPRO. O uso do Elasticsearch, além de conferir a escalabilidade necessária para o funcionamento do ENSEPRO com BC extensas, possibilitou uma simplificação das consultas, possibilitando a abstração da acentuação, da caixa alta e pesquisas por casamento parcial de palavras.

Como se pode perceber na Figura Figura 10, os processos de combinação e ranqueamento de candidatas à resposta foram implementados na linguagem de programação Java. Conforme visto nas seções 4.3.3 Geração combinatorial de candidatas à resposta e 4.3.4 Ranqueamento de respostas, o uso da linguagem Java deveu-se à necessidade de otimização do desempenho destes processos. A comunicação entre os processos em Python e Java foram implementadas através de troca de arquivos. O uso de word embedding no algoritmo de seleção de respostas do ENSEPRO se dá por uma chamada de API externa para verificação de similaridade entre as palavras. A API foi desenvolvida visando a abstração do algoritmo de avaliação de similaridade. No protótipo atual, optou-se pelo uso da biblioteca de código aberto gensim[32] para a implementação deste serviço.

O sistema é desenvolvido em um computador com sistema operacional Windows 10, mas testado e homologado em ambiente Linux virtualizado em um contêiner Docker. O Quadro 34 apresenta a ficha técnica dos principais componentes dos ambientes de desenvolvimento e homologação do protótipo.

[27] https://github.com/dbpedia-spotlight/spotlight-docker
[28] http://compling.hss.ntu.edu.sg/omw
[29] http://www.nltk.org/howto/wordnet.html
[30] https://www.elastic.co/pt/products/elasticsearch
[31] https://lucene.apache.org
[32] https://radimrehurek.com/gensim/models/keyedvectors.html

Quadro 34: Ficha técnica dos ambientes de desenvolvimento e homologação do ENSEPRO.

1. Ambiente de Desenvolvimento e Teste
    1. Hardware
        1. Computador Intel i7
        2. Memória RAM 8 GB
        3. HD 240 GB
    2. Software
        1. Sistema Operacional: Windows 10
        2. Editor de Desenvolvimento
        3. Python 3.6
        4. Java Oracle 1.8
2. Ambiente de Homologação
    1. Hardware
        1. Servidor Dell PowerEdge R630
        2. CPU Intel(R) Xeon(R) v4 de 2,1 GHz de 32 núcleos
        3. Memória RAM 128 GB
        4. Disco Dell SCSI de 600 GB
    2. Software
        1. Sistema operacional do servidor Ubuntu 16.04.6 LTS
        2. Docker versão 18.09.5-0ubuntu1~16.04.2
3. Linguagens de Programação e principais bibliotecas
        1. Python 3.6
        2. NLTK 3.2.5
        3. Gensim 3.7.0
        4. Java OpenJDK 1.8.0_181
        5. Elasticsearch 6.2.4
        6. Spotlight 1.0

Como o sistema ENSEPRO envolve muitos processos e APIs de comunicação, a lista completa de softwares e dependências necessárias para a sua execução é bastante longa. Uma documentação dos sistemas necessários para a execução do sistema foi disponibilizada nos seguintes repositórios públicos do projeto ENSEPRO no Github[33]:

1. ensepro-core: código principal do ENSEPRO em Python.
2. ensepro-answer-generator: implementação em Java do módulo de combinação e ranqueamento de candidatas a resposta.
3. ensepro-similarity-service: implementação em Python do serviço de similaridade de palavras via word embedding.
4. ensepro-palavras-service: API de abstração do parser linguísticos
5. ensepro-dbpedia-dataset-to-elasticsearch: código em Python para carga da BC no Elasticsearch
6. ensepro-experiments: scripts bash e código Python utilizados para execução dos experimentos de avaliação do ENSEPRO.

[33] https://github.com/Ensepro

5.1.1 Pré-processamento da Base de Conhecimento

Sendo o objetivo deste trabalho desenvolver um sistema de PRS, é imprescindível atender requisitos inerentes a este tipo de aplicação. Dentre os principais requisitos dos sistemas de PRS destaca-se o tempo de resposta.

Tem-se como objetivo que o sistema ENSEPRO possa ser utilizado com BC relativamente extensas. Para tanto, é imprescindível que o acesso às informações seja o mais otimizado possível, uma vez que o tempo de acesso à BC é um dos fatores determinantes para o tempo de resposta do sistema.

Neste contexto, decidiu-se pela adoção de um sistema para otimizar o tempo de acesso à BC, uma vez que os sistemas tradicionais de gerenciamento de acesso à BC não apresentaram desempenho apropriado. Neste contexto, optou-se pelo uso do Elasticsearch[34] como sistema para otimização do tempo de busca para o ENSEPRO.

O Elasticsearch é um motor de busca full-text (BLAIR, 1984) distribuído de alta escalabilidade. O uso do Elasticsearch para otimização do acesso é uma abordagem comumente adotada em se tratando de otimização de acesso a bases de dados muito extensas (KONONENKO et al., 2014). No Quadro 35 pode-se ver a estrutura de dados utilizada para o armazenamento da BC.

Quadro 35: Estrutura do índice utilizado para armazenar a BC no Elasticsearch.

1. Triple
   1.1. subject
      1.1.1. original_text
      1.1.2. URI
      1.1.3. concept
   1.2. predicate
      1.2.1. original_text
      1.2.2. URI
      1.2.3. concept
   1.3. object
      1.3.1. original_text
      1.3.2. URI
      1.3.3. concept

A estrutura de dados criada para armazenamento da BC no Elasticsearch compõe-se de três elementos principais: subject (item 1.1), predicate (item 1.2) e object (item 1.3). Cada um destes elementos contém 3 subcampos: (1) original_text, campo para armazenamento dos dados originais da tripla; (2) URI, é um campo preenchido com a IRI do elemento da tripla; e, por fim, no campo (3) concept será armazenada a parte que representa o conceito da IRI ou o valor da propriedade de dados, no caso de literais.

No Quadro 36 apresenta-se um exemplo de como uma tripla é armazenada no Elasticsearch. O campo original_text (linhas 3, 7 e 11) tem como objetivo armazenar os dados originais da tripla, pois são realizadas algumas transformações antes de armazená-los no Elasticsearch, como a substituição dos caracteres acentuados por seus equivalentes sem

[34] https://www.elastic.co/products/elasticsearch

acento (“ã” por “a” e “ó” por “o”) e a padronização de todos os carácteres em caixa baixa. O objetivo destas transformações é normalizar os dados de forma a possibilitar consultas mais eficientes.

Quadro 36: Tripla armazenada no Elasticsearch.

```
1. Dados armazenados no Elasticsearch:
2.  “subject”: {
3.      “original_text”: “<http://pt.DBPedia.org/resource/Japão>”,
4.      “URI”: “http://pt.DBPedia.org/resource/”,
5.      “concept”: “japao”}
6. “predicate”: {
7.      “original_text”: “<http://pt.DBPedia.org/property/capital>”,
8.      “URI”: “http://pt.DBPedia.org/property/”
9.      “concept”: “capital”}
10. “object”: {
11.     “original_text”: “<http://pt.DBPedia.org/resource/Tóquio>”,
12.     “URI”: “http://pt.DBPedia.org/resource/”,
13.     “concept”: “toquio”}
```

Os campos URI (linhas 4, 8 e 12) e concept (linhas 5, 9 e 13) contém os dados normalizados da tripla. O motivo de armazenar estes campos separadamente é possibilitar que o módulo CBC possa recuperar as triplas que casem parcialmente com o campo concept.

Todos os exemplos apresentados neste capítulo fazem referência sempre ao conteúdo do campo concept, uma vez que as buscas realizadas pelo sistema ENSEPRO são realizadas sobre este campo.

6 AVALIAÇÃO DE RESULTADOS

Este capítulo tem por objetivo apresentar a avaliação do trabalho desenvolvido no decorrer da implementação do projeto, tanto para verificar-se o desempenho da abordagem proposta, quanto para explicitar-se as contribuições científicas da pesquisa desenvolvida.

Além de apresentar os resultados obtidos no experimento de avaliação do sistema ENSEPRO em si, considerou-se importante mostrar neste capítulo o desempenho dos algoritmos implementados para classificação de tipos questão e de seleção de TR do módulo CLN, uma vez que ambos sugerem novas abordagens para a realização das suas atividades.

Para a realização da avaliação do desempenho de sistemas de PRS são necessários pelo menos dois artefatos: um corpus contendo as perguntas e as suas respectivas respostas e a BC para a realização da busca das respostas. Após levantamento realizado no intuito de localizar-se estes dois artefatos para a avaliação do sistema ENSEPRO, concluiu-se não haver ainda disponível um recurso que permitisse avaliar um sistema de PRS que tenha por objetivo o processamento de perguntas elaboradas em português.

Neste sentido, julgou-se importante empenhar esforços na implementação de um artefato que pudesse ser utilizado para avaliação do sistema ENSEPRO, mas que pudesse também ser reutilizado pela comunidade científica para avaliação do desempenho de outros sistemas de PRS cujo o foco seja processar perguntas em português.

A partir deste contexto, considerou-se importante relatar-se também neste capítulo a metodologia adotada para a confecção do corpus elaborado para a realização do experimento de avaliação do sistema ENSEPRO, uma vez que a disponibilização deste artefato é uma das contribuições científicas deste projeto de pesquisa.

Uma vez detalhados os procedimentos adotados para a criação dos corpus de avaliação, são apresentados os resultados obtidos pelo sistema ENSEPRO em dois experimentos de avaliação. Primeiramente apresenta-se o desempenho em relação ao corpus do QALD-7 adaptado à DBPedia em português. Em seguida são apresentados os resultados obtidos com a aplicação do sistema ENSEPRO para responder perguntas sobre informações armazenadas em uma ontologia da área da saúde. Por fim, faz-se o fechamento do capítulo apresentando a análise dos resultados obtidos.

6.1 Avaliação da classificação do tipo de questão

Para que um sistema de PR possa responder a perguntas formuladas em linguagem natural é necessário um algoritmo que implemente algum nível de compreensão da questão recebida. Na abordagem aqui proposta, a implementação do algoritmo de compreensão decorre das informações linguísticas da frase em linguagem natural.

Conforme visto na seção 4.2 A Compreensão da Linguagem Natural, são geradas informações complementares da frase, como a voz do verbo (ativa ou passiva), os termos relevantes (substantivos e verbos), a presença de locuções verbais e de complementos nominais, bem como o tipo de pergunta.

A classificação do tipo da pergunta é essencial para a implementação do algoritmo de compreensão do que o usuário deseja saber. A classificação do tipo de pergunta adotada neste trabalho é realizada por um algoritmo próprio que baseia-se nas informações linguísticas da pergunta para classificar o tipo da questão.

Para verificar o desempenho do classificador de tipo de pergunta realizou-se um experimento de avaliação. Para esta avaliação utilizou-se um corpus de 130 questões contendo um total de 1007 palavras, com perguntas de 4 a 17 palavras no máximo. As frases do corpus foram manualmente pré-classificadas quanto ao tipo de pergunta conforme a Figura Tabela 1 apresentada na seção 4.2.3 Classificação do tipo de questão.

Para o cálculo de desempenho geral do classificador de tipos de perguntas foi utilizada a avaliação de classificação multiclasses baseada na média do desempenho obtido para cada tipo de pergunta (HOSSIN; SULAIMAN, 2015). Primeiramente verificou-se a precisão (1), a revocação (2) e a acurácia (3) para cada tipo de pergunta presente no corpus de avaliação. A Tabela Tabela 10 apresenta os resultados obtidos nesta primeira avaliação.

Precisão $\frac{tp}{tp+fp}$ (1)

Revocação $\frac{tp}{tp+fn}$ (2)

Acurácia $\frac{tp+tn}{tp+fn+fp+tn}$ (3)

Tabela 10: Desempenho do classificador por tipo de pergunta.

Métrica	Algum	Quem	O que	Consulta	Onde	Alguém	Qual	Média
Precisão	***1,00***	***1,00***	***0,25***	***1,00***	***1,00***	***1,00***	***1,00***	***0,89***
Revocação	***1,00***	***1,00***	***1,00***	***0,96***	***1,00***	***1,00***	***0,98***	***0,99***
Acurácia	***1,00***	***1,00***	***0,98***	***0,98***	***1,00***	***1,00***	***0,99***	***0,99***

Para a verificação do desempenho geral do classificador de tipo de pergunta foram calculadas as métricas de Precisãoμ, Revocaçãoμ, Acuráciaμ e Escore F1μ do algoritmo, utilizando-se respectivamente as fórmulas 4, 5, 6 e 7. Os resultados obtidos nesta avaliação são apresentados na Tabela Tabela 11.

$$\text{Precisão}_{\mu} \quad \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fp_i)} \tag{4}$$

$$\text{Revocação}_{\mu} \quad \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fn_i)} \tag{5}$$

$$\text{Acurácia}_{\mu} \quad \frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \tag{6}$$

$$\text{Escore F1}_{\mu} \quad 2 \times \frac{precisão \times revocação}{precisão + revocação} \tag{7}$$

Tabela 11: Desempenho geral do sistema de classificação de tipos de pergunta.

	Precisão$_{\mu}$	**Revocação$_{\mu}$**	**Acurácia$_{\mu}$**	**Escore F1$_{\mu}$**
Média	0,9769	0,9769	0,9934	0,9769

Percebe-se que o classificador implementado teve um bom desempenho na definição do tipo de pergunta. Credita-se o bom resultado alcançado à opção de utilizar as informações linguísticas fornecidas pelo parser. A partir das informações linguísticas das frases é possível identificar-se muito precisamente o tipo da pergunta, utilizando-se um algoritmo de comparação baseado nas características linguísticas fornecidas pelo parser, conforme visto na seção 4.2.3 Classificação do tipo de questão.

O funcionamento do algoritmo de comparação é favorecido por um procedimento chamado de Mecanismo de Atenção, o qual remove de todas as palavras da frase que não tem relevância para a classificação do tipo de pergunta, utilizando como base a própria lista de tipos. Por ser baseado em regras, o algoritmo de classificação não depende de treinamento prévio. Pode-se perceber na descrição do algoritmo que o seu desempenho é explicitamente baseado nas informações geradas pelo parser linguístico.

Embora o algoritmo desenvolvido tenha apresentado um desempenho bastante satisfatório na classificação das questões, a dependência do desempenho do parser pode ser considerada como um ponto sensível da abordagem. Para que o classificador apresente bons resultados é crucial a adoção de um parser com desempenho compatível com o estado da arte.

6.2 Avaliação da identificação de Complementos Nominais

Além da classificação do tipo de pergunta, outra atividade muito importante para a localização das respostas é a identificação dos complementos nominais presentes na frase, conforme visto na seção 4.2.4.2 Tratamento dos Complementos Nominais. O algoritmo de

identificação de CNs do sistema ENSEPRO é também baseado em regras e foi implementado com base nas frases do mesmo corpus de 130 perguntas utilizado na avaliação do algoritmo de classificação do tipo de pergunta.

Para a avaliação da identificação de CNs buscou-se um novo corpus de perguntas contendo 14.178 palavras distribuídas em 1.452 questões[35] de domínio aberto. O algoritmo de identificação foi aplicado sobre este corpus, tendo sido então manualmente verificadas todas as frases e identificadas falhas na identificação de CN em 261 perguntas.

Os 261 erros foram analisados individualmente, concluindo-se que somente ocorreram falhas na localização dos CNs específicos. Ou seja, a precisão alcançada pelo algoritmo foi de 100%, pois o algoritmo acertou todas as vezes que indicou a presença de um CN. Já a revocação, devido aos 261 erros apontados, foi de 98,16%. Os erros ocorridos deveram-se principalmente a falta de exemplos do corpus de 129 perguntas, além de 19 casos em que o parser falhou em sua análise devido a erros gramaticais nas frases.

A partir dos resultados acima, analisou-se os casos em que houve erro de identificação, concluindo-se ser possível implementar as mudanças necessárias no algoritmo para a correta identificação dos casos faltantes. A partir da implementação das mudanças no código do algoritmo, rodou-se novamente o experimento sobre o corpus, desta vez corrigindo-se os erros gramaticais das 19 perguntas.

Para este novo experimento obteve-se uma precisão e revocação perfeitas (100%). Se por um lado percebe-se que o algoritmo de identificação de CNs, assim como o de Classificação do Tipo de Pergunta, também depende dos resultados retornados pelo parser, por outro pode-se perceber que o conjunto de regras implementados a partir de 129 perguntas teve um excelente nível de generalização para o processamento do corpus de 1.452 questões.

Concluiu-se que os desempenhos obtidos no classificador de tipos de perguntas e de identificação de CNs são decorrência direta da qualidade e do nível de abstração das informações fornecidas pelo parser linguístico, pois as etiquetas morfossintáticas e a árvore de dependências geradas permitiram a implementação dos respectivos sistemas de regras com desempenho bastante expressivos.

Os bons resultados obtidos, tanto na classificação de tipo de pergunta (visto na seção anterior) quanto na identificação de CNs, são fundamentais para o desempenho alcançado pelo processo de seleção de palavras chaves do ENSEPRO, assunto da próxima seção.

6.3 Avaliação do processo de seleção de palavras chaves

Para avaliar o desempenho do algoritmo para identificação de palavras-chave, utilizou-se como padrão ouro o conjunto de dados de treinamento QALD-7-multilingual, disponibilizado no 7º Question Answering Linked Data Challenge (USBECK et al., 2017).

O conjunto de dados QALD-7-multilingual usado neste experimento de avaliação é um arquivo no formato JSON contendo questões traduzidas em nove idiomas, sendo um deles o português brasileiro (linhas 7 e 8 do Quadro 37). Para cada idioma em que a frase foi traduzida, além de outros dados relacionados ao desafio em si, são dadas as respectivas palavras-chaves de cada questão (linhas 4 e 9 da Quadro 37).

[35] Disponível em https://www.tediado.com.br/10/556-perguntas-amigos-ask/ - visitado em 05/05/2019.

Quadro 37: Trecho do arquivo QALD-7-multilingual.json em contendo uma pergunta e respectivas palavras-chave em inglês e português.

```
1.  "question": [{
2.    "language": "en",
3.    "string": "Who was the wife of U.S. president Lincoln?",
4.    "keywords": "U.S. president, Lincoln, wife"
5.  },
6.  {
7.    "language": "pt_BR",
8.    "string": "Quem foi a esposa do presidente americano Lincoln?",
9.    "keywords": "esposa, presidente americano, Lincoln"
10. }]
```

O experimento para avaliar a identificação de palavras-chave consistiu na submissão das sentenças em português do QALD-7 para o ENSEPRO e, para fins de comparação, também a um algoritmo baseado em TF-IDF (RAMOS, 2003). Depois da submissão, são comparadas as palavras-chave apontadas pelos sistemas em relação às palavras-chaves listadas no item keywords para a versão em língua portuguesa (linha 9 do Quadro 37) do conjunto de dados do QALD-7.

O arquivo QALD-7-multilingual possui 215 sentenças em português brasileiro, totalizando 1553 palavras (média de 7,22 palavras/frase), das quais 787 são identificadas como palavras-chave. O algoritmo de seleção de palavras-chave do sistema ENSEPRO identificou corretamente 784 palavras-chave (verdadeiro positivo – VP), 748 palavras comuns (verdadeiro negativo – VN), classificando erroneamente 18 palavras comuns como palavras-chave (falso positivo – FP), não encontrando 3 palavras-chave (falso negativo – FN).

$$\text{Escore F1} \qquad \frac{TP+TN}{TP+TN+FP+FN} \qquad (8)$$

O método TF-IDF foi treinado e testado com as mesmas 215 frases em português brasileiro do QALD-7, identificando corretamente 779 palavras-chave e 666 palavras comuns, classificando erroneamente 98 palavras comuns como palavras-chave, não encontrando 10 palavras-chave. Com base nesse desempenho, calculou-se as métricas de precisão (1), revocação (2), acurácia (3) e escore F1 (8). Os desempenho de ambos os sistemas (ENSEPRO e TF-IDF) são mostrados na Tabela 12.

Tabela 12: Desempenhos obtidos pelo algoritmo de seleção de palavras chaves do sistema ENSEPRO comparado a um método baseado em TF-IDF.

Abordagem	Precisão	Revocação	Acurácia	Escore F1
ENSEPRO	0.9776	0.9962	0.9865	0.9868
TF-IDF	0.8883	0.9873	0.8782	0.9352

Como o algoritmo de seleção de palavras-chave do ENSEPRO é baseado em regras, não é necessário haver uma etapa de treinamento anterior para ser usado. Já o método baseado em TF-IDF demanda uma etapa de treinamento anterior ao uso. A fim de garantir-se uma comparação adequada, a abordagem baseada em TF-IDF foi treinada e testada com o mesmo conjunto de dados.

Procedeu-se dessa forma para garantir-se que o método usado como base de comparação conhecesse previamente todas as palavras a serem processadas durante a fase de teste. Como podemos ver na Tabela Tabela 12, mesmo usando os mesmos dados para treinamento e teste do método TF-IDF, ainda assim o algoritmo do sistema ENSEPRO apresentou melhores resultados em todas as métricas.

Atribui-se os melhores resultados ao fato de que a abordagem do sistema ENSEPRO usa informações linguísticas de um nível de abstração mais alto que o método TF-IDF. Assim, mesmo sendo um método baseado em regras estáticas, nossa abordagem é mais genérica porque usa as informações morfossintáticas e estruturais do analisador linguístico para decidir se uma palavra é ou não uma palavra-chave. O uso de informações linguísticas de um nível de abstração mais elevado possibilita a definição de regras com generalidade suficiente para compensar o fato de serem estáticas.

Embora os resultados tenham sido considerados bastante satisfatórios, realizou-se ainda uma análise individual dos erros para compreender melhor as suas origens. Nessa análise, identificamos dez causas que impediram nossa abordagem de identificar as palavras-chave definidas no conjunto de dados do QALD-7. Na Tabela Tabela 13 apresenta-se um exemplo de frase para ilustrar cada uma dessas causas.

Tabela 13: Diferenças (destacadas em negrito) entre as palavras-chaves apontadas pelo QALD-7-multilingual e as selecionadas pelo algoritmo do sistema ENSEPRO.

Sentença	QALD-7	ENSEPRO
1) Em que cidade fica a cervejaria Heineken?	cidade, cervejaria Heineken	cervejaria Heineken
2) O Príncipe Harry e o Príncipe William têm os mesmos pais?	Príncipe Harry, Príncipe William, mesmos pais	Príncipe Harry, Príncipe William, pais
3) Quantas pessoas vivem na Polônia?	quantas pessoas, Polônia	pessoas, vivem, Polônia
4) A Crise dos Mísseis de Cuba ocorreu antes da Invasão da Baía dos Porcos?	Crise, Mísseis, Cuba, antes, Invasão, Baía dos Porcos	Crise, Mísseis, Cuba, ocorreu, Invasão, Baía dos Porcos
5) Qual é o nome verdadeiro do Batman?	nome, verdadeiro, Batman	nome, Batman
6) O local onde Abraham Lincoln morreu tem um web site?	local, Abraham Lincoln, morreu, web site	Abraham Lincoln, morreu, web site
7) Liste todas as aves que correm perigo de extinção.	aves, perigo, extinção	aves, correm, perigo, extinção
8) Com quem Lance Bass se casou?	esposa, Lance Bass	Lance Bass, casou

Sentença	QALD-7	ENSEPRO
9) Qual cidade tem menos habitantes?	cidade, menos habitantes	cidade, habitantes
10) Kaurismäki alguma vez venceu o Grand Prix de Cannes?	Kaurismäki, vencedor, Grand Prix de Cannes	Kaurismäki, vez, vencedor, Grand Prix de Cannes

Na frase 1 da Tabela Tabela 13 o termo “cidade” não foi identificado como uma palavra-chave devido a forma pela qual trabalham o sistema de classificação de frases e o algoritmo de identificação de palavras-chaves do ENSEPRO: primeiramente realiza-se a classificação do tipo de frase e posteriormente a identificação das palavras-chave. A busca pelas palavras-chaves inicia após o último termo que identifica o tipo de frase.

Na época em que o experimento de avaliação foi realizado, a sequência “em que cidade” era utilizada para identificar o tipo de questão chamado de “localização”. A partir dos erros observados neste experimento, verificou-se que não é uma boa prática definir as sequências para identificação do tipo de frase com substantivos. Ao analisar trechos da frase como indicadores do seu tipo, o ENSEPRO ignora as palavras que compõe este trecho no momento de selecionar as palavras-chave da frase.

Em uma análise mais aprofundada desta questão, verificou-se que o problema em relação a considerar trechos da frase como indicativos do seu tipo prejudica a interpretação da frase quando palavras deste trecho são substantivos. Dependendo do significado do substantivo na frase, a sua omissão na lista de TRs pode até mesmo inviabilizar o algoritmo de localização da resposta do ENSEPRO. A partir destas constatações, revisou-se a lista de tipos de frases, removendo-se todos os tipos cuja a definição do tipo contivesse substantivos.

Em relação à divergência na frase 2, avaliou-se que o sistema ENSEPRO está correto em não considerar a palavra “mesmos” como um termo relevante para a busca da resposta, ao menos não em BC semânticas. A palavra “mesmos” tem sim um significado em relação à consulta, pois a sua presença na frase indica que duas consultas devem ser feitas e comparadas as respostas.

No caso de BC semânticas, é pouco provável que a tripla contendo a resposta para a questão contenha uma tripla do tipo “[Príncipe_Harry mesmosPais Príncipe_William]”. A probabilidade maior é que ocorram triplas representando as relações de paternidade e maternidade dos indivíduos, as quais deveriam então ser recuperadas da BC e comparadas para então responder-se a pergunta recebida.

No caso da pergunta da frase 2, o termo “mesmos” é um indicativo de que deve-se fazer uma consulta para cada um dos substantivos próprios da frase (Príncipe Harry e Príncipe William) e comparar-se a igualdade dos resultados obtidos. Dado este contexto, percebe-se que a palavra “mesmos” não é relevante para a realização da consulta em si, pois para esta frase a palavra “mesmos” não vai aparecer nas triplas necessárias para responder a questão.

No caso da frase 3 existem duas divergências: a palavra “quantas” e para a palavra “vivem”. A palavra “quantas” foi ignorada por ser considerada como uma marca do tipo de pergunta (mesmo caso da palavra “cidade” na frase 1). Diferentemente do caso da frase 1, para a frase 3 acredita-se que houve um equívoco ao classificar-se “quantas” como uma

palavra-chave pois, além do termo não ser relevante para a localização da resposta por não aparecer na tripla que responde a questão, a palavra “quantas” não é classificada como chave em outras 14 perguntas em que aparece do corpus.

Em relação à classificação do termo “vivem” como palavra-chave na frase 3, o algoritmo de seleção do ENSEPRO a classificou como chave porque ela é um verbo principal e relacional. No que se refere ao sistema ENSEPRO, verbos relacionais da pergunta sempre são considerados como termos relevantes da frase. Embora no caso específico da frase 3 seja pouco provável que a palavra “viver” seja utilizado na BC como um predicado para representar o número de habitantes de um local, na abordagem aqui apresentada propõe-se que serão buscadas na BC as triplas que contém tanto as palavras-chaves quanto os seus sinônimos.

Especificamente em relação à análise do termo “viver” ter sido considerado como uma palavra-chave da frase 3, é razoável considerar-se que “viver” seja um sinônimo de “residir”, o qual por sua vez estaria relacionado com a população de um determinado local. Embora no caso específico da frase 3 a interpretação do verbo “viver” como “população” seja bastante mais complexo, pois está associado ao uso do termo (ou seja, o significado da palavra está no nível pragmático da linguística), é importante considerar que há uma grande probabilidade de que os verbos relacionais citados explicitamente nas perguntas componham as triplas da resposta.

Sendo então um fator que interfere diretamente na eficiência do sistema em localizar respostas na BC, não se considerou como um erro a classificação da palavra “vivem” como uma palavra-chave. Analisando-se cada uma das frases em que houve divergência entre o QALD-7 e o ENSEPRO em relação aos verbos (frases 1, 3, 4, 7 e 8 da Tabela 1), concluiu-se que a seleção dos verbos é fator decisivo para o desempenho do sistema e, a partir disto, decidiu-se que, mesmo divergindo do padrão adotado pelos autores do corpus, a lógica de sempre considerar os verbos relacionais como palavras-chaves seria mantida.

Em relação às frases 4, 5, 8, 9 e 10, as divergências devem-se ao fato de que o ENSEPRO considera como palavras chaves os termos que são substantivos ou verbos (salvo as exceções apontadas na seção 4.2.4 Seleção dos Termos Relevantes). Todas as divergências em relação às palavras-chaves do QALD-7 para estas frases estão relacionadas ao algoritmo de seleção implementado no ENSEPRO.

Não se considerou um erro não classificar como chaves as palavras “antes” (frase 4) e “menos” (frase 9), pois no que se refere a busca da resposta na BC, estas palavras não seriam relevantes na consulta à BC. Também não foi considerado erro o caso da frase 8, pois “esposa” é inserido como palavra-chave mas sequer aparece na pergunta.

A divergência em relação à classificação de “verdadeiro” (frase 5) como palavra-chave remete à questão de considerar-se ou não adjetivos como palavras-chaves. Esta questão, conforme análise realizada na seção 4.3.4.6 Quantidade de referências a adjetivos na tripla candidata, foi resolvida da seguinte forma: o adjetivo é importante para a seleção da resposta, mas não para a consulta na BC. Sendo assim, os adjetivos não são palavras-chaves no momento da consulta à BC, mas são considerados no momento de definir o valor de ranque da tripla.

Outro caso interessante a comentar é a palavra “vez” da frase 10, a qual foi considerada pelo ENSEPRO como sendo um termo relevante, o que efetivamente não é

positivo, pois esta palavra não irá compor as triplas e portanto não contribuirá para a localização da resposta na BC. Até onde pode-se observar nos experimentos realizados, não é totalmente negativo o impacto de haver substantivos que são classificados como TRs mas que efetivamente não contribuem para a localização da resposta. O algoritmo de busca das respostas implementado no módulo CBC tem a robustez suficiente para assimilar a presença destes "falsos TRs".

Em relação às diferenças entre o QALD-7 e o ENSEPRO, seriam estas as situações que mereceram destaque. Dados os bons desempenhos observados na avaliação, considera-se que as divergências devem-se ora a uma certa incoerência observada no QALD-7 em relação aos critérios para seleção das palavras-chaves, ora devido à rigidez das regras do ENSEPRO. Contudo, observa-se pelo desempenho alcançado no experimento de avaliação que o algoritmo para seleção de palavras-chaves implementado apresenta eficiência adequada às necessidades do sistema ENSEPRO.

6.4 Criação do corpus QALD-7 em Português Brasileiro

A metodologia científica para a Ciência da Computação, especialmente nos trabalhos relacionados à Computação Aplicada, pressupõe uma avaliação objetiva do artefato computacional resultante da pesquisa, seja frente a um conjunto padrão de resultados esperados (o chamado padrão ouro), seja via comparação direta de desempenho com outros sistemas desenvolvidos (WAZLAWICK, 2015). No caso do desenvolvimento de sistemas de PRS, é igualmente importante que os artefatos computacionais estejam disponíveis para permitir a comparação de desempenho entre as diferentes abordagens desenvolvidas. Conjuntos de dados como o SimpleQuestion (BORDES et al., 2015) e o WebQuestion (BERANT et al., 2013) são recursos que possibilitam essa comparação objetiva. No entanto, esses artefatos estão disponíveis apenas para a língua inglesa.

Com relação aos recursos computacionais para avaliar sistemas PRS, tanto quanto se pode-se levantar em pesquisa bibliográfica, o único recurso publicamente disponível para outros idiomas além do inglês é o desafio QALD (UNGER et al., 2014). No entanto, embora o QALD tenha incluído a tarefa multilíngue desde sua terceira edição (CIMIANO et al., 2013), somente na edição de 2018 do evento o conjunto de dados recebeu a inclusão de questões em português[36]. No entanto, analisando as 549 questões dos dados de treinamento e teste disponíveis[37], pode-se concluir que tradutores automáticos foram utilizados para gerar as questões em português. Essa abordagem comprometeu significativamente a qualidade das sentenças geradas. A maior parte das traduções das questões para o português são de difícil compreensão até mesmo para falantes nativos.

A escassez de recursos para comparar as abordagens compromete a avaliação de sistemas de PRS em português, dificultando o estabelecimento do estado da arte para esses sistemas. Consequentemente, também é difícil validar a evolução da pesquisa nessa área. Se, por um lado, existe um número expressivo e crescente de sistemas PRS atualmente visando a língua inglesa, por outro lado, existe uma situação muito diferente para a língua portuguesa, com poucos trabalhos publicados. Em relação aos procedimentos de avaliação adotados,

[36] https://project-hobbit.eu/challenges/qald-9-challenge/
[37] https://github.com/ag-sc/QALD/tree/master/9/data

observa-se a falta de um padrão objetivo de comparação do desempenho dos sistemas SQA em português nos trabalhos publicados. Devido ao esforço necessário para produzir um corpus para avaliação de desempenho, geralmente os trabalhos são avaliadas com algumas poucas dezenas de perguntas (KETSMUR; RODRIGUES; TEIXEIRA, 2017), o que compromete não só a avaliação da capacidade de generalização da abordagem, mas também a comparação com outros sistemas.

De forma a contribuir e estimular o desenvolvimento de procedimentos de avaliação mais abrangentes e aprofundados para os sistemas de PRS em português, decidiu-se fazer os esforços necessários para desenvolver e disponibilizar um corpus em português que permita realizar a avaliação do desempenho dos sistemas. Nas seções seguintes, descrevemos nossa abordagem, desde a escolha do corpus original até as estratégias adotadas na elaboração do artefato.

6.4.1 Metodologia adotada

Quanto aos procedimentos gerais seguidos para a criação do corpus de avaliação do sistema ENSEPRO, pode-se dizer que o trabalho foi desenvolvido em três etapas. O primeiro passo foi o estudo de estrutura de dados dos arquivos disponibilizados pelo QALD. O corpus do QALD compõe-se fundamentalmente de arquivos contendo questões visando a avaliação de desempenho de sistemas de PR.

No momento em que o trabalho de criação do corpus de avaliação estava em andamento, o conjunto de dados mais atual disponibilizado era o QALD-7, sendo este então o conjunto de dados escolhido como base para a criação do corpus de avaliação, mais especificamente o QALD-7-multilingual.

Quadro 38: Trecho do arquivo JSON do QALD-7-multilingual contendo as questões e respectivas palavras-chaves em várias línguas, a consulta SPARQL e a resposta para a questão.

```
"question": [
   {"language": "en",
   "string": "When was the Battle of Gettysburg?",
   "keywords": "Battle of Gettysburg"},
   {"language": "de",
   "string": "Wann fand die Schlacht von Gettysburg statt?",
   "keywords": "Schlacht von Gettysburg"},  ...
      ],
"query": {
   "sparql": "     PREFIX dbo: <http://DBPedia.org/ontology/>
              PREFIX res:<http://DBPedia.org/resource/>
              SELECT DISTINCT ?date
              WHERE {res:Battle_of_Gettysburg dbo:date ?date .\n}"},
"answers": [{"head": { "vars":  [ "date" ] },
   "results": {
         "bindings": [{"date": {"type": "literal", "value": "1863-07-03"}]
      }
}, ...
```

O segundo passo foi fazer a tradução das perguntas e das respectivas palavras-chaves. Cada questão do QALD-7-multilingual é composta pela definição do idioma, a pergunta em linguagem natural, as respectivas palavras-chaves, a consulta SPARQL para localização da resposta e as triplas resultantes da execução da consulta. Um exemplo da estrutura de dados utilizada para representar as questões pode ser visto no Quadro 38.

Por fim, na terceira e última etapa, criou-se um segundo corpus de avaliação a partir da adaptação das consultas SPARQL e respostas do QALD-7-multilingual à versão em língua portuguesa da DBPedia (DBPedia PT). No primeiro conjunto de dados disponibilizado, as consultas e respostas não foram alteradas, mantendo-se a referência original à DBPedia em inglês. Já no segundo conjunto de dados foram alteradas as consultas SPARQL e as respostas em relação ao conteúdo da DBPedia PT.

6.4.2 Tradução das questões e respectivas palavras-chaves

O primeiro procedimento para traduzir as perguntas do QALD-7-multilingual foi reunir e analisar as 258 questões e adotar um conjunto de procedimentos a observar na tradução. Decidimos que a tradução consideraria o português falado no Brasil, uma vez que o grupo de pesquisa envolvido neste trabalho não possui falantes nativos de português de Portugal.

A tradução das perguntas foi realizada concomitantemente por dois membros do grupo de pesquisa: uma linguista com especialização em inglês e um pesquisador em computação com especialização em Processamento de Linguagem Natural.

A tradução das perguntas foi realizada de forma independente pelos dois pesquisadores e, em seguida, os resultados foram combinados em uma única lista. Esse procedimento foi adotado com a intenção de evitar a propensão natural dos profissionais de cada área de criar uma tradução tendenciosa. As duas listas de traduções foram compiladas em uma única lista por acordo entre os dois pesquisadores.

Para o especialista em linguística foi fornecida apenas a lista de perguntas em inglês, pois esta era a língua em que o pesquisador tinha especialização e possuía extensa experiência em tradução de documentos. O especialista em computação trabalhou diretamente com o arquivo JSON e também teve acesso às demais traduções disponíveis no arquivo.

Houve divergências entre as duas traduções e a escolha de palavras para formular a pergunta, o que é bastante natural, dada a diferente formação de ambos os pesquisadores e a flexibilidade intrínseca da Linguagem Natural. Tais divergências foram discutidas e resolvidas buscando-se sempre a forma usual de formular a questão em português falado no Brasil e também utilizando as traduções já disponíveis no conjunto de dados para as outras línguas.

As principais divergências entre as traduções realizadas pelos dois pesquisadores ocorreram nas referências a entidades geopolíticas e obras cinematográficas. No Quadro 39 são apresentados alguns exemplos para ilustrar as situações mais típicas de divergência.

Quadro 39: Alguns exemplos de traduções em que houveram discordâncias e respectivo texto final, destacando-se em negrito as divergências.

Tradução originalmente proposta:

1. Quem foi a esposa do presidente **dos Estados Unidos** Lincoln?
   (Who was the wife of U.S. president Lincoln?)
2. Quais são os cinco condados de **Nova Iorque**?
   (What are the five boroughs of New York?)
3. Quem é o prefeito da capital da **Polinésia Francesa**?
   (Who is the mayor of the capital of French Polynesia?)
4. Quem é o prefeito de **Rotterdam**?
   (Who is the mayor of Rotterdam?)

Texto final:

1. Quem foi a esposa do presidente **americano** Lincoln?
2. Quais são os cinco condados de **New York**?
3. Quem é o prefeito da capital da **Polinésia Francesa**?
4. Quem é o prefeito de **Rotterdam**?

Na sentença 1 do Quadro 39, pode-se verificar que o nome do país (Estados Unidos) foi substituído pelo adjetivo geralmente adotado para se referir aos cidadãos daquele país (americano). A tradução proposta originalmente foi alterada devido à decisão de seguir o padrão adotado nas outras línguas românicas (POSNER, 1996) presentes no conjunto de dados: espanhol, francês, italiano e romeno. A decisão de buscar consonância com as línguas românicas assumidas neste trabalho baseia-se no fato de que a língua portuguesa faz parte desse conjunto de linguagens que evoluíram do latim, tendo por isso semelhanças quanto ao estilo e semântica da escrita (HALL, 1974).

Quadro 40: Traduções nas línguas românicas para a frase em inglês "Who was the wife of the American president Lincoln?".


```
{       "language": "en",
        "string": "Who was the wife of U.S. president Lincoln?",
        "keywords": "U.S. president, Lincoln, wife" },
{       "language": "pt_BR",
        "string": "Quem foi a esposa do presidente americano Lincoln?",
        "keywords": "esposa, presidente americano, Lincoln"},
{       "language": "es",
        "string": "¿Quién fué la mujer del presidente americano Lincoln?",
        "keywords": "presidente Americano, Lincoln, mujer"},
{       "language": "it",
        "string": "Chi era la moglie del presidente degli Stati Uniti Lincoln?",
        "keywords": "presidente degli Stati Uniti, Lincoln, moglie"},
{       "language": "fr",
        "string": "Qui était l'épouse du président américain Lincoln?",
        "keywords": "épouse, président américain Lincoln"}
```

Estabeleceu-se como procedimento para a tradução seguir o padrão adotado pela maioria das línguas românicas ou, em caso de divergência, observar o procedimento mais comumente adotado no português falado no Brasil. Analisando as traduções para as outras línguas românicas apresentadas no Quadro 40, constatou-se que a inclusão do nome do país em detrimento do gentílico ocorreu apenas na versão em italiano, motivo pelo qual modificou-se a tradução originalmente proposta, visando acompanhar o padrão adotado pela maioria das outras línguas românicas.

Embora as outras divergências apresentadas no Quadro 39 também se originem da aplicação da mesma regra (ou seja, seguir o padrão adotado pela maioria das línguas românicas), resultados diferentes são observados no texto final. O termo “Polinésia Francesa” na frase 3, por exemplo, permaneceu em português no texto final, enquanto ‘Nova Iorque” na frase 2 foi alterado para sua versão em inglês. Curiosamente, para a cidade de Rotterdam foi proposto pelo linguista utilizar-se o termo em inglês (sentença 4), o que foi mantido no texto final pela regra de consonância com as outras línguas românicas.

Em relação às diferenças de tradução para obras cinematográficas, pode-se ver pelo exemplo apresentado no Quadro 41 que algumas pequenas alterações superficiais foram feitas, mas que não alteram a essência da questão. Essas mudanças devem-se principalmente à regra de consonância com as línguas românicas. Em relação ao exemplo do Quadro 41, vale ressaltar que representa um desafio interessante para os pesquisadores da área de sistemas de PRS.

Geralmente os títulos em português atribuídos a obras cinematográficas não têm relação direta com o título original. Decorre deste costume que o nome de filmes de outros países podem ser completamente diferentes do título original. O exemplo apresentado no Quadro 41 é um caso típico da ocorrência desta situação. No Brasil e em Portugal este filme ganhou o título “Ensina-me a viver”, bastante diferente do título original “Harold and Maude”[38].

Quadro 41: Tradução proposta e texto final para a pergunta em inglês “Who composed the song for Harold and Maude?".

Tradução proposta:

1. Quem compôs a música para Harold e Maude?

Texto final

1. Quem é o compositor da música de Harold e Maude?

Embora o texto final da pergunta seja um desafio semântico bastante complexo para sistemas de PRS, para a escolha do texto final considerou-se o fato de que o objetivo da tarefa multilíngue do QALD é obter as respostas a partir da BC da DBPedia em inglês. Portanto, seria totalmente incoerente utilizar o título em português, uma vez que impossibilitaria a referência correta na DBPedia em inglês.

A partir desta situação, detectados outros casos semelhantes no conjunto de dados do QALD, decidiu-se pela produção de uma versão dos dados do QALD-7 específico para a DBPedia em português. Os desafios enfrentados nesta empreitada de produzir este novo corpus estão descritos na 6.4.3 Adaptação das respostas do QALD-7 à DBPedia PT.

[38] https://www.imdb.com/title/tt0067185

Acredita-se que com estes procedimentos obteve-se uma tradução, tanto quanto possível, isenta da influência de conceitos preconcebidos dos tradutores. Após concluir os procedimentos de tradução das perguntas, iniciamos o trabalho de seleção das palavras-chave das frases em língua portuguesa.

Para a seleção de palavras-chave, considerou-se que era necessário conhecimento específico na área de Recuperação de Informações. Por esse motivo, a etapa relacionada à elaboração das palavras-chaves foi realizada apenas pelo pesquisador com especialização em Extração de Informações. Definiu-se que para esta etapa seriam utilizados novamente o padrão de seleção adotado pelas outras línguas românicas presentes no conjunto de dados. Talvez por ser um procedimento mais objetivo, raramente houve casos de discordância entre as línguas românicas na seleção de palavras-chave.

Nos raros casos em que houve discordância, foi realizada uma análise mais abrangente, envolvendo também os idiomas inglês e alemão. Pode-se ver no Quadro 42 um exemplo típico em que foi necessária uma análise mais aprofundada. Constatou-se para esta frase três procedimentos diferentes para a seleção de palavras-chave nas línguas românicas: (1) restringir-se ao uso de palavras que estão contidas na sentença ignorando o pronome interrogativo (espanhol); (2) inserção de palavras não encontradas na frase original (italiano e francês); e (3) uso de palavras contidas na pergunta, incluindo o pronome interrogativo (romeno).

Quadro 42: Palavras-chaves das línguas românicas para a pergunta "Quem é o compositor da música de Harold e Maude?".

```
"question": [
        {       "language": "en",
                "string": "When did Operation Overlord commence?",
                "keywords": "when, Operation Overlord, commence" },
        {       "language": "pt_BR",
                "string": "Quando começou a Operação Overlord?",
                "keywords": "começou, Operação Overlord"},
        {       "language": "es",
                "string": "¿Cuándo comenzó la operación Overlord?",
                "keywords": "comienzo, operación Overlord "},
        {       "language": "it",
                "string": "Quando è iniziata l'operatione Overlord?",
                "keywords": "data di inizio, operatione Overlord"},
        {       "language": "fr",
                "string": "Quand a commencé l'opération Overlord?",
                "keywords": "date de commencement, opération Overlord"},
        {       "language": "ro",
                "string": "Când a început operațiunea Overlord?",
                "keywords": "când, început, operațiunea Overlord"}
], ...
```

Para casos como esse apresentado no Quadro 42 decidiu-se adotar o seguinte padrão de procedimentos: primeiro, seguir o procedimento definido de seleção de palavras-chave

observando para a língua portuguesa os procedimentos adotados nas línguas românicas mais próximas do português, seguindo esta ordem de influência: espanhol, italiano, francês e romeno. Assim, para o caso de divergência apresentado no Quadro 42, definiu-se que as palavras-chave para a frase em português seria “começou” e “Operação Overlord”, influenciado pela seleção de palavras adotadas na língua espanhola, mas restrito à forma lexical original da palavra na frase, na qual se pode observar o verbo conjugado na primeira pessoa do singular do pretérito perfeito.

É importante mencionar que o procedimento para a seleção das palavras-chave descrito acima foi seguido sem exceções, independentemente de concordar-se ou não com as escolhas feitas. Por exemplo: o procedimento observado para a maioria das questões foi desconsiderar o pronome interrogativo como palavra-chave. No entanto, como se vê no Quadro 43, por alguma razão desconhecida, este padrão não foi seguido. Como o pronome foi incluído como palavra-chave em todas as línguas românicas, seguiu-se este procedimento também para o português, ainda que se considere um tanto incoerente este procedimento.

Quadro 43: Pronome interrogativo incluído nas palavras-chaves das línguas românicas para a pergunta “Quantas pessoas vivem na Polônia?"


```
"question": [
        {
                "language": "en",
                "string": "How many people live in Poland?",
                "keywords": "How many people, Poland"},
        {
                "language": "pt_BR",
                "string": "Quantas pessoas vivem na Polônia?",
                "keywords": "quantas pessoas, Polônia"},
        {
                "language": "es",
                "string": "¿Cuántas personas viven en Polonia?",
                "keywords": "Cuántas personas, habitan, Polonia"},
        {
                "language": "it",
                "string": "Quante persone vivono in Polonia?",
                "keywords": "quante persone, Polonia"},
        {
                "language": "fr",
                "string": "Combien de personnes vivent en Pologne?",
                "keywords": "nombre de personnnes, Pologne"},
        {
                "language": "ro",
                "string": "Câți oameni locuiesc în Polonia?",
                "keywords": "câți oameni, Polonia"}
], ...
```


Embora não exista uma regra explícita para definir quais palavras devem ser consideradas chaves no contexto da Recuperação da Informação, é bastante claro que o

pronome interrogativo é raramente incluído como uma palavra-chave nas outras sentenças do QALD-7. Então, unicamente para fins de consistência de procedimento, inseriu-se a palavra "quantas", de acordo com o padrão da maioria das outras línguas românicas.

Após a seleção das palavras-chave para as frases em português, deu-se por finalizado os trabalhos relativos a tradução das frases para o português falado no Brasil. Contatou-se então os responsáveis pela organização do QALD e colocou-se à disposição o trabalho desenvolvido. A iniciativa foi bem recebida pelos organizadores do evento que passaram orientações sobre os procedimentos para integrar o trabalho desenvolvido ao repositório oficial do QALD-7[39].

Como citado anteriormente, no decorrer do trabalho de tradução das frases decidiu-se adaptar as respostas do conjunto de dados do QALD-7 à DBPedia PT. A próxima seção descreve detalhes e dificuldades enfrentadas no decorrer deste empreitada.

6.4.3 Adaptação das respostas do QALD-7 à DBPedia PT

O processo de adaptar o conjunto de dados do QALD-7 envolveu a análise individual de cada uma das 258 perguntas do QALD-7 para identificar as mudanças que seriam necessárias no conjunto de dados. O procedimento de análise começou identificando as mudanças na consulta SPARQL para cada questão. Esta etapa foi necessária para identificar os elementos do DBPedia EN referenciados na consulta para encontrar os elementos equivalentes no DBPedia PT. No Quadro 44 apresenta-se um exemplo das diferenças entre as consultas SPARQL e as respostas encontradas na DBPedia EN e DBPedia PT para a pergunta "Quem é o prefeito de Rotterdam?".

Quadro 44: Diferenças entre as consultas e respostas da DBPedia EN e da DBPedia PT.

1. Questão:

Quem é o prefeito de Rotterdam?

2. Consulta SPARQL para a DBPedia EN:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT DISTINCT ?uri
WHERE { res:Rotterdam dbo:leaderName ?uri. }
```

3. Resposta DBPedia EN:

```
"uri": {
"type": "uri",
"value": "http://dbpedia.org/resource/Ahmed_Aboutaleb"
}
```

4. Consulta SPARQL para a DBPedia PT:

```
PREFIX dbp: <http://pt.dbpedia.org/property/>
PREFIX res: <http://pt.dbpedia.org/resource/>
SELECT ?s WHERE { res:Roterdão dbp:líderNome ?s .  }
```

5. Resposta DBPedia PT:

```
"s": { "type": "literal",  "value": "Michel Buillard" }
```

[39] https://github.com/ag-sc/QALD/tree/master/7/data

Primeiramente, observe-se as diferenças entre as consultas SPARQL para a localização da resposta (itens 2 e 4 do Quadro 44). Como os projetos DBPedia EN e DBPedia PT possuem autonomia para definir as suas estruturas de dados, pode-se perceber diferenças quanto ao vocabulário e até mesmo diferenças estruturais e referenciais na representação da informação.

Cada DBPedia escolhe autonomamente os detalhes do seu vocabulário. No exemplo, podemos ver essas diferenças em relação aos termos “Rotterdam / Roterdão” e “leaderName / líderNome”. Também pode-se perceber as diferenças referenciais de representação da informação. A DBPedia EN faz referência à URI http://dbpedia.org, enquanto a DBPedia PT utiliza a UTI http://pt.dbpedia.org como base de referência.

Em termos estruturais, isso demonstra que cada projeto definiu diferentes domínios para a relação entre a cidade e o nome de seu prefeito. A DBPedia EN associou a esta relação um conceito da ontologia (http://dbpedia.org/ontology), enquanto a DBPedia PT definiu-a mais especificamente como uma propriedade da ontologia (http://pt.dbpedia.org/property).

Mas não são apenas as consultas que precisaram ser alteradas para a adaptação do conjunto de dados ao DBPedia PT. Também as respostas demandam mudanças, como pode ser visto nos itens 3 e 5 do Quadro 44. Para a DBPedia EN a resposta é uma URI, enquanto que para a DBPedia PT a resposta é um literal. Desconsiderando as questões referenciais desta diferença, em termos do padrão de representação adotado no QALD JSON, essa diferença no tipo de resposta exige mudanças mesmo no nome das variáveis da consulta (“?uri” na DBPedia EN e “?s” na DBPedia PT).

No decorrer do trabalho de adaptação do banco de dados à DBPedia PT, observou-se que as diferenças entre as DBPedias estão presentes em tal extensão que todas as consultas SPARQL sofreram alterações. O exemplo apresentado no Quadro 44 representa os casos mais simples da adaptação feita. Houve casos muito mais complexos que levaram um tempo considerável para serem implementados. As dificuldades e contratempos enfrentados, as estratégias e procedimentos realizados para a realização da atividade, bem como as lições aprendidas, são detalhadamente descritas nas próximas seções.

6.4.4 Adaptações necessárias

Após a migração das 258 perguntas do QALD-7 para a DBPedia PT, percebeu-se que foram desenvolvidos um conjunto de estratégias para fazer as adaptações necessárias. Imediatamente após a conclusão do trabalho, fez-se a catalogação e investigação dessas estratégias, a fim de compartilhar os procedimentos adotados e as lições aprendidas. Identificou-se cinco procedimentos para construção da consulta SPARQL: (1) resposta simples, (2) resposta dependente da categoria, (3) resposta incorreta e (4) pergunta não respondida.

O primeiro procedimento é construir a nova consulta SPARQL. Durante a construção da consulta, identifica-se o tipo de resposta a ser representada no arquivo JSON. A partir do tipo de resposta, identifica-se o tipo de procedimento a ser seguido para inserir a resposta no arquivo JSON. Cada um dos procedimentos para representar a resposta, embora definido no decorrer da análise, teve como resultado principal uma implementação padronizada e coerente. Entende-se que este é um dos requisitos fundamentais de um conjunto de dados para

avaliação de sistemas. As seções a seguir descrevem em detalhes cada um dos processos envolvidos na adaptação das respostas do QALD-7 ao DBPedia PT.

6.4.5 A construção da consulta SPARQL

O procedimento de adaptação inicia pela construção da consulta SPARQL equivalente para DBPedia PT a partir da análise da consulta original do QALD. Nesta análise busca-se identificar primeiro as referências aos nomes próprios, ou seja, identificar os elementos da consulta que referem-se aos nomes próprios presentes na questão.

Por exemplo, para a pergunta "Quem é o prefeito de Paris?" (Item 1 do Quadro 45), primeiramente identifica-se qual elemento da consulta à DBPedia EN (item 2) se refere ao nome Paris. Podemos então verificar que na consulta a DBPedia EN é a URI <http://dbpedia.org/resource/Paris> que se refere ao nome Paris.

Quadro 45: Consulta SPARQL e respectiva resposta nas DBPedias EN e PT para a pergunta "Quem é o prefeito de Paris?".

1. Questão:

Quem é o prefeito de Paris?

2. Consulta SPARQL para a DBPedia EN:

```
SELECT DISTINCT ?uri
WHERE {  <http://dbpedia.org/resource/Paris> <http://dbpedia.org/ontology/mayor> ?uri
.
}
```

3. Consulta SPARQL para a DBPedia PT:

```
SELECT ?uri
WHERE {<http://pt.dbpedia.org/resource/Paris> <http://pt.dbpedia.org/property/prefeito>
?uri .
}
```

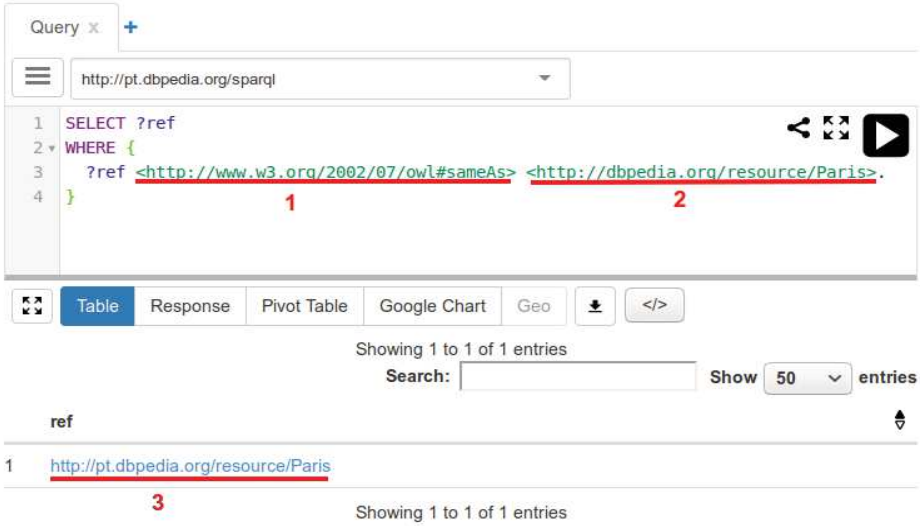
O processo para descobrir as referências equivalentes na DBPedia PT é bastante simples, como pode ser visto na Figura 11. Consiste em consultar a DBPedia PT usando a propriedade sameAs do OWL (1) e a referência original do DBPedia EN (2), sendo então retornada a referência equivalente na DBPedia PT (3). Para essas consultas utilizou-se o YASGUI[40], um cliente SPARQL disponível na Web (RIETVELD; HOEKSTRA, 2017), o qual disponibiliza uma interface de usuário muito intuitiva e simplificada.

Quando uma referência equivalente não é encontrada, isso significa que a pergunta do QALD-7 não tem resposta na DBPedia PT. Mais adiante apresenta-se o procedimento adotado para os casos de perguntas sem resposta, pois existem algumas outras situações que resultam em perguntas sem resposta.

Prosseguindo com o exemplo: quando a referência equivalente for encontrada, então torna-se necessário identificar a relação equivalente na DBPedia PT, isto é, identificar qual predicado usado na DBPedia PT é equivalente ao predicado da DBPedia EN.

[40] http://yasgui.org/

Figura 11: Consulta na BDPedia PT pela referência equivalente à DBPedia EN na YASGUI.



Fonte: elaborada pelo autor.

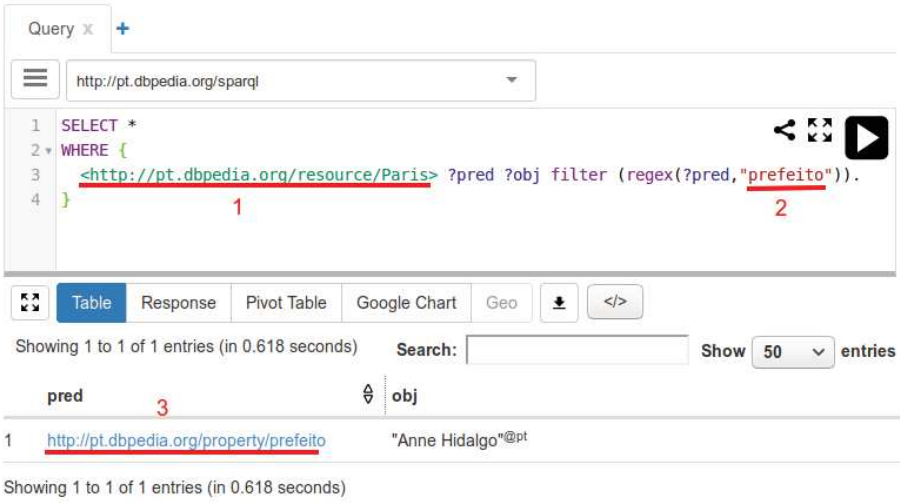
Esse é um processo mais complicado, porque os predicados não seguem um padrão como no caso das referências. No caso da questão apresentada no Quadro 45, por exemplo, podemos ver que o predicado da DBPedia EN <http://dbpedia.org/ontology/mayor> (item 2 do Quadro 45) não está relacionado ao predicado utilizado na DBPedia PT <http://pt.dbpedia.org/property/prefeito> (item 3). Eles não têm apenas o URI base diferente (http://dbpedia.org e http://pt.dbpedia.org), mas também diferentes domínios (ontology e property).

O procedimento inicial é a verificação da existência de um predicado na DBPedia PT que contém em sua URI uma tradução para o português do termo utilizado em inglês. Na Figura Figura 12 vê-se um exemplo desse procedimento para o predicado <http://dbpedia.org/ontology/mayor> da DBPedia EN.

O primeiro passo é listar todas as triplas da DBPedia PT com a referência equivalente <http://pt.dbpedia.org/resource/Paris> (item 1) e a tradução para o português do termo principal como predicado (item 2). O resultado da consulta retorna que o predicado equivalente no DBPedia PT é <http://pt.dbpedia.org/property/prefeito> (item 3).

Se o procedimento de busca pelo predicado equivalente pela tradução falhar, realiza-se uma nova consulta no DBPedia PT, desta vez com a referência equivalente e o predicado originalmente utilizado no DBPedia EN. Por exemplo, se a consulta mostrada na Figura Figura 12 não tivesse retornado nenhuma resposta, seria realizada uma nova consulta no DBPedia PT, desta vez usando a referência equivalente <http://pt.dbpedia.org/resource/Paris> como subject da tripla e o predicado original da DBPedia EN <http://dbpedia.org/ontology/mayor>.

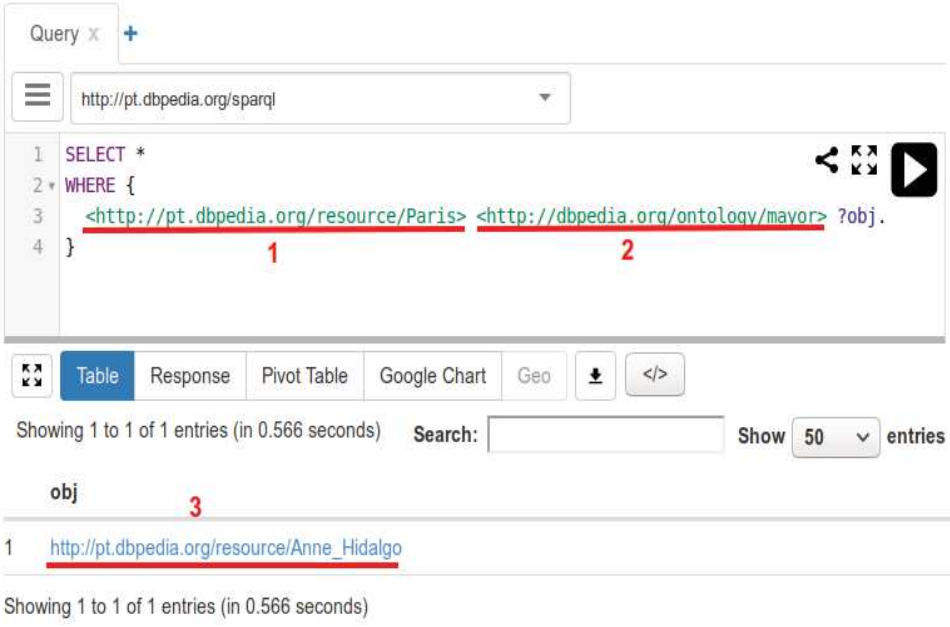
Figura 12: Busca do predicado equivalente na DBPedia PT pela tradução do predicado original da DBPedia EN.



Fonte: elaborada pelo autor.

Pode-se ver na Figura Figura 13 que a consulta da referência equivalente (1) combinada com o predicado original da DBPedia EN (2) também retorna um resultado (3), o qual é diferente do resultado da consulta por tradução do predicado. Pode-se observar que o resultado retornado (<http://pt.dbpedia.org/resource/Anne_Hidalgo>) é estruturalmente mais semelhante ao retorno da consulta na DBPedia EN.

Figura 13: Consulta à DBPedia PT utilizando a referência equivalente e o predicado original da consulta à DBPedia EN.



Fonte: elaborada pelo autor.

Realiza-se sempre primeiro a busca do predicado equivalente pela tradução porque entende-se que esta é a abordagem mais adequada para a produção de um conjunto de dados para a validação de sistemas SQA para português, uma vez que procedendo assim dá-se preferência ao uso do predicado expresso em português.

No entanto, os procedimentos descritos acima nem sempre resultam na localização de um predicado equivalente. Se ambos os procedimentos falharem, faz-se uma tentativa mais genérica de localizar o predicado equivalente: análise de todas as triplas do DBPedia PT que contêm a referência equivalente, filtrando-se progressivamente as triplas até que o predicado equivalente desejado seja encontrado. Este procedimento é um processo de exploração manual que consome um tempo considerável em alguns casos mais complicados ocorridos no decorrer do trabalho de adaptação do QALD-7 para a DBPedia PT.

Não descreve-se aqui tais procedimentos porque entendemos que eles não são reutilizáveis, uma vez que esses casos não resultaram em um procedimento padrão a ser seguido, mas em um conjunto muito específico de ações, que dependem de minúcias características de cada situação encontrada.

Novamente, se o predicado equivalente não for encontrado, significa que a questão QALD-7 não pode ser respondida com base nas informações representadas no DBPedia PT, resultando em uma questão sem resposta. O tratamento das questões sem resposta será detalhado em uma seção específica mais adiante. Se o processo de construção da nova consulta SPARQL for bem-sucedido, ou seja, todas as referências e predicados equivalentes são encontrados, podemos prosseguir para o procedimento final: analisar as respostas encontradas e fazer as alterações no arquivo JSON. Identificou-se quatro situações diferentes em relação às respostas, cada qual com o seu conjunto específico de procedimentos. As seções a seguir descrevem cada uma dessas situações.

6.4.5.1 Resposta simples

Se, após o processo de construção da nova consulta SPARQL, obtém-se no DBPedia PT um retorno que pode ser considerado como a resposta correta para a questão, tem-se então o caso de simples inserção da resposta no JSON. Chama-se de resposta simples até mesmo as respostas que não são compatíveis com a resposta original, desde que a informação retornada seja a resposta correta.

Quadro 46: Exemplo de resposta simples com tipos compatíveis, destacando-se em negrito os trechos em que houveram alterações.

1. Questão:
Quanto custou Pulp Fiction?

2. JSON file with SPARQL query and original answer from DBPedia EN:

```
"query": {"sparql": "SELECT DISTINCT ?n WHERE {
                        <http://dbpedia.org/resource/Pulp_Fiction>
                        <http://dbpedia.org/ontology/budget>
                        ?n . } " },
"answers": [ {
      "head": { "vars": [ "n" ] },
      "results": {  "bindings": [ { "n": { "type": "literal", "value": "8.0" } }]      }
} ]
```

3. JSON file with SPARQL query and adapted answer from DBPedia PT:

```
"query": { "sparql": "SELECT DISTINCT ?n WHERE {
                        <http://pt.dbpedia.org/resource/Pulp_Fiction>
                        <http://pt.dbpedia.org/property/orçamento>
                        ?n . } "},
"answers": [ {
      "head": { "vars": [ "n" ] },
      "results": {  "bindings": [ { "n": { "type": "literal", "value": "US$ 8 milhões" } }] }
} ]
```

O exemplo apresentado no Quadro 46 é um caso de resposta simples com tipo compatível, uma vez que tanto o DBPedia EN quanto o DBPedia PT possuem respostas do tipo literal. É interessante notar que o conteúdo retornado pelas consultas no item value são diferentes. Mesmo retornando respostas diferentes, devido à compatibilidade de tipos, considera-se este exemplo como sendo um caso de resposta simples.

O procedimento para respostas simples consiste em apenas atualizar o JSON da resposta original (item 2 do Quadro 46), substituindo o conteúdo do item sparql e valor da resposta. O item 3 do Quadro 46 mostra como ficaram as informações do JSON do DBPedia PT. Pode haver casos em que o DBPedia PT tenha uma resposta correta cujo tipo é diferente do tipo original. Considera-se que esse tipo de situação também é uma resposta simples, pois a única diferença em relação ao exemplo apresentado acima é que é necessário atualizar o campo type, o que o caracteriza como um caso simples de ser resolvido.

6.4.5.2 Respostas dependentes de categorias

Uma situação muito mais complexa a resolver é o caso de respostas dependentes da categoria da Wikipédia. Para encontrar a resposta para esse tipo de pergunta do conjunto de dados do QALD-7 foi necessário examinar-se como são organizados os elementos que

compõem uma categoria de artigos. A análise do exemplo apresentado no Quadro 47 ilustra detalhes do contexto deste tipo de questão.

Quadro 47: Trecho do arquivo JSON do QALD-7 contendo a consulta SPARQL e as respostas para uma pergunta dependente de categoria na DBPedia EN.

1. Questão:

Give me all American presidents in the last 20 years.

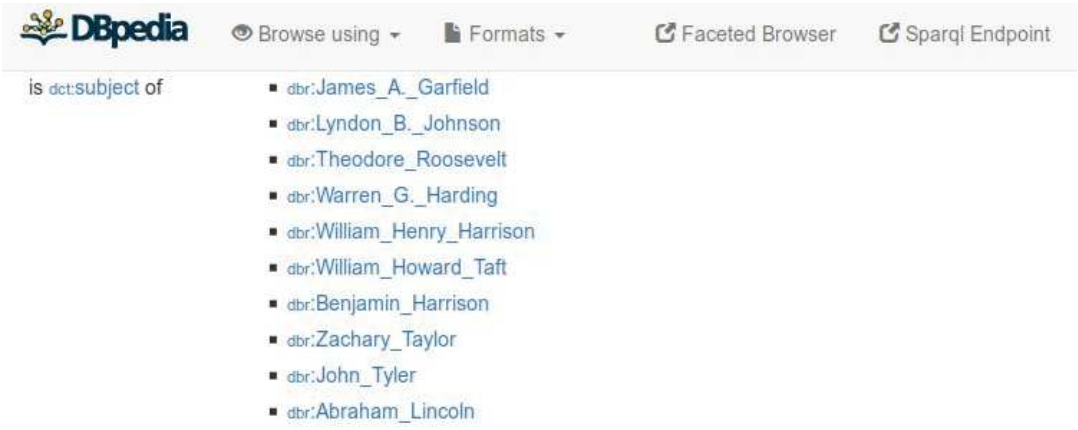
2. consulta SPARQL e respostas na DBPedia EN:

```
{"query": {  "sparql": "
            PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
            PREFIX dbo: <http://dbpedia.org/ontology/>
            PREFIX dct: <http://purl.org/dc/terms/>
            PREFIX dbc: <http://dbpedia.org/resource/Category:>
            PREFIX dbp: <http://dbpedia.org/property/>
            select distinct ?uri where {
                  ?uri rdf:type dbo:Person .
                  ?uri dct:subject dbc:Presidents_of_the_United_States .
                  ?uri dbp:termEnd ?termEnd .
            FILTER ( year( NOW() ) - year( ?termEnd ) <= 20 ) }"
 },
{"answers": [ {  "head": { "vars": [ "uri" ] },
{"results": {  "bindings": [
            { "uri": { "type": "uri", "value": "http://dbpedia.org/resource/Bill_Clinton"}},
            { "uri": { "type": "uri", "value":
"http://dbpedia.org/resource/George_W._Bush"}},
            { "uri": { "type": "uri", "value": "http://dbpedia.org/resource/Barack_Obama" }
      }
      ]
}
```

As questões dependentes de categoria são facilmente identificáveis pela presença na consulta SPARQL de uma referência a URI <http://dbpedia.org/resource/Category:>. Na consulta SPARQL da questão apresentada no Quadro 47, por exemplo, percebe-se a referência à URI de categoria <http://dbpedia.org/resource/Category:Presidents_of_the_United_States>.

Uma forma simples de visualizar essas URIs de categoria é carregá-las diretamente em um navegador web, já que as informações relacionadas à categoria são apresentadas em formato HTML (Figura 14). Dentre as várias informações que aparecem na DBPedia EN relacionadas a URI de categoria <http://dbpedia.org/resource/Category:Presidents_of_the_United_States>, pode-se ver uma lista contendo os presidentes daquele país.

Figura 14: Página visualizada em navegador web ao carregar-se diretamente a URI de categoria <http://dbpedia.org/resource/Category:Presidents_of_the_United_States>.



Fonte: elaborada pelo autor.

Para retornar a informação solicitada na pergunta “Give me all American presidents in the last 20 years.” do Quadro 47 é necessário verificar o ano de mandato de todos os presidentes dos EUA, o que é realizado pela verificação de todas as triplas que têm relação através do predicado <http://purl.org/dc/ terms/subject> com a URI <dbc:Presidents_of_the_United_States>.

A adaptação da consulta SPARQL para perguntas dependentes de categoria passa pela tradução das URIs para o DBPedia PT. Primeiramente é necessária a substituição da URI por <http://pt.dbpedia.org/resource/Categoria:> e um pequeno ajuste na especificação do filtro devido à diferença no tipo da variável “termEnd”, como pode ser visto no Quadro 48.

Embora as etapas para inserir a resposta no JSON sejam semelhantes às respostas simples, é importante criar uma categoria específica para apresentar este caso, já que o procedimento para localizar referências e propriedades equivalentes é específico neste tipo de questão. Não há, tanto quanto se verificou, uma sequência definida de procedimentos a seguir para a localização das URIs equivalentes.

Verificou-se nas questões envolvendo respostas dependentes de categoria que não há um procedimento único para a modelagem da representação das categorias na DBPedia PT, o que impossibilitou a definição de um procedimento de adaptação das URIs de referência. Se por um lado é bastante simples identificar que a resposta é dependente de categoria pela verificação da presença da URI <http://dbpedia.org/resource/Category:> na consulta SPARQL da DBPedia EN, por outro lado o processo de elaboração da consulta à DBPedia PT é geralmente muito trabalhoso e demorado.

Em relação às questões dependentes da categoria, pode-se perceber no exemplo apresentado que houve uma diferença nas respostas de cada DBPedia. Observa-se na lista de respostas que, diferentemente do retorno obtido na consulta à DBPedia EN (item 2 do Quadro 47), ao executar a consulta SPARQL da DBPedia PT (item 2 do Quadro 48), não há uma URI de referência ao ex-presidente Bill Clinton, um problema provavelmente causado por algum erro no algoritmo de geração da BC da DBPedia PT. Este é um exemplo em que a resposta

encontrada na DBPedia PT não está totalmente correta, assunto este que será visto na próxima seção.

Quadro 48: Trecho do arquivo JSON do QALD-7 contendo a consulta SPARQL e as respostas para uma pergunta dependente de categoria na DBPedia PT.

1. Questão:

Me dê todos os presidentes americanos dos últimos 20 anos.

2. consulta SPARQL e respostas na DBPedia PT:

```
{"query": {  "sparql": "
            PREFIX dbo: <http://dbpedia.org/ontology/> PREFIX dct:
<http://purl.org/dc/terms/>
            PREFIX dbc: <http://pt.dbpedia.org/resource/Categoria:>
            select distinct ?uri where {
                  ?uri <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> dbo:Person .
                  ?uri dct:subject dbc: Presidentes_dos_Estados_Unidos .
                  ?uri <http://pt.dbpedia.org/property/anos> ?termEnd .
            FILTER ( year( NOW() ) - ?termEnd  <= 20 ) }"},
{"answers": [ {  "head": { "vars": [ "uri" ] },
{"results": {  "bindings": [
            { "uri": { "type": "uri", "value":
"http://pt.dbpedia.org/resource/George_W._Bush"}},
            { "uri": { "type": "uri", "value": "http://pt.dbpedia.org/resource/Barack_Obama"
}}]
      }
}
```

6.4.5.3 Respostas incorretas

O fato de receber respostas incorretas ou incompletas ocorreu em diversas situações durante o trabalho de adaptação das respostas do QALD-7 à DBPedia PT. Fez-se um levantamento das principais divergências entre as duas bases de conhecimento e verificou-se a presença de informação incorreta ora na base da DBPedia PT, ora na DBPedia EN. A origem dos erros nas respostas das DBPedias em relação às questões do QALD-7 tem relação tanto com dados errados armazenados na base quanto com a falta de artigos da Wikipédia que permitissem responder corretamente a questão.

Dada este contexto, concluiu-se que a melhor abordagem para produzir o conjunto de dados para a avaliação de sistemas de PRS seria considerar como correto o retorno obtido a partir da consulta da base de conhecimento, ainda que a resposta não fosse correta. No Quadro 49 apresenta-se um exemplo de uma questão em que se considera correto que o sistema PRS retorne apenas à URI <http://pt.dbpedia.org/resource/Pinta_(caravela)> como resposta, já que este é o retorno possível com as informações armazenadas na DBPedia PT.

Quadro 49: Trecho do arquivo JSON do QALD-7 contendo a consulta SPARQL e as respostas para uma pergunta com resposta incompleta na DBPedia PT.

1. Questão:

Quais eram os nomes dos três navios usados por Colombo?

2. consulta SPARQL e respectiva resposta na DBPedia PT:

```
"query": {
        "sparql":"
                PREFIX dbo: <http://dbpedia.org/ontology/>
                PREFIX dbc: <http://pt.dbpedia.org/resource/Categoria:>
                select ?uri where {
                        ?uri  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  dbo:Ship.
                        ?uri  <http://purl.org/dc/terms/subject> dbc:Cristóvão_Colombo.}"
                },
"answers": [{
        "head": { "vars": ["uri"] },
        "results": { "bindings": [
                { "uri": {"type": "uri",
"value":"http://pt.dbpedia.org/resource/Pinta_(caravela)"} }
                ] }
        }]
```

Acredita-se ser este é o procedimento mais coerente com os objetivos do trabalho, afinal o sistema de PRS somente pode retornar como resposta aquilo que está efetivamente representado na base de conhecimento. Se a informação na BC está errada ou incompleta, mas é a resposta possível a ser encontrada para a questão, considera-se que o sistema de PRS cumpriu com sucesso a sua função.

Outra situação que enfrentada em relação à adaptação dos corpus do QALD-7 à DBPedia PT refere-se a questões cujas respostas não estão representadas na BC. Este tópico é o assunto da próxima seção.

6.4.5.4 Resposta inexistente

O objetivo original do corpus disponibilizado no QALD-7 é possibilitar a avaliação do desempenho dos sistemas de PRS com base nas informações representadas na DBPedia EN. Como todas as questões do corpus foram formuladas considerando o conteúdo da DBPedia EN, algumas delas estão relacionadas a informações que não estão representadas na DBPedia PT, sendo então impossível obter-se a resposta.

Um exemplo de pergunta que não pode ser respondida consultando-se a DBPedia PT é apresentado no Quadro 50. A pergunta não pode ser respondida porque não há informação alguma sobre a pessoa em questão na BC da DBPedia PT. Como não há uma consulta a ser feita, devido à inexistência de dados na BC, decidiu-se que as questões seriam representadas no corpus com os campos “query” e “answers” vazios (item 2 do Quadro 50).

Quadro 50: Exemplo de uma questão com resposta inexistente na DBPedia PT.

1. Questão:
Qual a altura de Amazon Eve?
2. Conteúdo do arquivo JSON adaptado à DBPedia PT para essa pergunta:
"query": {},
"answers": []

A decisão de como representar as perguntas sem resposta do corpus adaptado à DBPedia PT inspirou-se na forma pela qual foram representadas as perguntas sem resposta do corpus original do QALD-7. No Quadro 51 apresenta-se um exemplo de pergunta originalmente sem resposta do QALD-7. Se executarmos a consulta SPARQL usando o comando SELECT não haverá um retorno, já que não há uma tripla que corresponda à pesquisa feita.

Quadro 51: Exemplo de uma questão com resposta inexistente na DBPedia PT.

1. Questão:
James Bond é casado?
2. Conteúdo do arquivo JSON original para essa pergunta:

```
"query": {
    "sparql": "
        PREFIX dbo: <http://dbpedia.org/ontology/>
        PREFIX res: <http://dbpedia.org/resource/>
        ASK WHERE { res:James_Bond dbo:spouse ?uri .}"
    },
"answers": [{ "head": {}, "results": {}, "boolean": false }]
```

Percebeu-se que todas as perguntas não respondidas eram do tipo booleano. Todas as perguntas não-booleanas do QALD-7 foram formuladas de forma que a resposta fosse encontrada na DBPedia EN. Concluiu-se que seria interessante avaliar a capacidade dos sistemas de PRS de concluir que a questão não poderia ser respondida com as informações armazenadas na base de conhecimento. Assim, decidiu-se que as questões sem resposta seriam mantidas no corpus adaptado.

O fato de não haver uma resposta é representado no corpus original do QALD-7 pela definição de campos “head” e “results” vazios. Na adaptação do corpus do QALD-7 à DBPedia PT decidiu-se que as perguntas sem respostas seriam marcadas de uma forma diferenciada, possibilitando aos usuários do corpus decidirem se fazem uso ou não destas questões.

A marca que diferencia as perguntas sem resposta do corpus adaptado são os campos “query” e “answer” vazios, pois nenhuma questão do corpus original tem estes campos vazios, nem mesmo as perguntas sem resposta. Este é o único caso em que não seguiu-se os padrões de representação estabelecidos no conjunto de dados original do QALD-7, uma vez que representar perguntas não-booleanas com respostas inexistentes tornou-se um diferencial específico do nosso trabalho. Este trabalho está disponível publicamente em https://github.com/DenisAraujo68/ensepro-qald-7_pt-br.

6.5 Experimento de avaliação com DBPedia em Português

Finalizados os trabalhos de confecção dos dois corpus de avaliação de sistemas de PRS em português, deu-se início aos experimentos de avaliação. Considera-se este experimento relatado aqui como a principal de avaliação desta pesquisa, pois avalia o sistema ENSEPRO processando perguntas em português e buscando as respostas em BC estruturada com vocabulário também em português.

Tendo por objetivo facilitar a fluência da leitura desta seção, repete-se sucintamente aqui as principais informações do corpus utilizado para realização do experimento de avaliação do desempenho do sistema ENSEPRO. No que se relaciona ao experimento realizado, o corpus compõe-se fundamentalmente de um arquivo JSON utilizado no desafio QALD-7 contendo 215 perguntas curtas expressas em português, contendo um total de 1.553 palavras, com frases variando de 3 até 14 palavras, 7 palavras em média e moda de 6 palavras e desvio padrão de 2,09.

Conforme comentado na seção 6.4.3 Adaptação das respostas do QALD-7 à DBPedia PT, foram produzidos dois corpus de avaliação, um que restringe-se a traduzir as perguntas e palavras chaves do QALD-7 e outro que adapta as respostas à versão 2016-10 em português da DBPedia[41] (DBPedia PT). São relatados nesta seção os resultados obtidos pelo sistema ENSEPRO frente a estes dois corpus de avaliação.

O experimento foi realizado utilizando-se um servidor Dell PowerEdge R630 com uma CPU Intel(R) Xeon(R) v4 de 2,1 GHz de 32 núcleos com 128 GB de memória RAM e um disco Dell SCSI de 600 GB, com sistema operacional Ubuntu 16.04.6 LTS. O protótipo foi executado utilizando-se 3 contêineres Docker com funcionalidades específicas, conforme apresentado na Tabela 14.

Tabela 14: Lista de contêineres Docker utilizados para a realização do experimento de avaliação.

Imagem Docker	**Função**
docker.elastic.co/elasticsearch/elasticsearch:6.2.4	Armazenamento da BC
dbpedia/spotlight-portuguese	Hospedar o sistema REM
denis/ensepro:v0.4	Hospedar o sistema ENSEPRO

Os contêineres para armazenamento da BC e hospedagem do sistema REM estão disponíveis publicamente no Docker Hub[42]. O contêiner utilizado pelo sistema ENSEPRO, embora não esteja disponibilizado publicamente, é um contêiner baseado na imagem Docker pública ubuntu:16.04 contendo o sistema ENSEPRO instalado, o qual está publicamente disponível no Github[43].

O procedimento de realização do experimento baseou-se na execução de um script que primeiramente extrai as perguntas em português do corpus de avaliação, as submetendo

[41] Disponível em https://wiki.dbpedia.org/downloads-2016-10.
[42] https://hub.docker.com
[43] Disponível em https://github.com/Ensepro

individualmente ao sistema ENSEPRO para coleta das respectivas respostas. As respostas são então comparadas com as respostas do corpus de avaliação, realizando-se uma análise booleana da igualdade entre as respostas. Em outras palavras, ou as respostas são iguais ou diferentes (verdadeiro ou falso), não havendo níveis intermediários de semelhança.

Seguiu-se este mesmo procedimento em ambas as avaliações do sistema ENSEPRO, ou seja, tanto em relação ao corpus do QALD-7 com respostas baseadas na DBPedia EN quando ao corpus com respostas da DBPedia PT. Na Tabela 15 apresenta-se os resultados obtidos no experimento realizado, bem como resultados obtidos por outros sistemas frente ao corpus original do QALD-7 (USBECK et al., 2017).

Tabela 15: Comparativo do desempenho dos sistemas de PRS participantes da tarefa 1 do QALD-7 e os resultados obtidos pelo sistema ENSEPRO no experimento de avaliação.

Métricas	**ganswer2**	**WDAqua**	**ENSEPRO (PT)**
Precisão Micro	0,113	-	0,662
Revocação Micro	0,561	-	0,487
Escore F1 Micro	0,189	-	0,561
Precisão Macro	0,557	0,490	0,605
Revocação Macro	0,592	0,540	0,597
Escore F1 Macro	0,556	0,510	0,593

Adotaram-se os mesmos procedimentos de avaliação e as mesmas fórmulas de cálculo do desafio QALD-7 (USBECK et al., 2017). Os valores apresentados na Tabela 15 mostram o desempenho do sistema ENSEPRO em relação à precisão e revocação de cada questão (q) segundo as seguintes fórmulas:

$$revocação(q) = \frac{número\ de\ respostas\ corretas\ do\ sistema\ para\ q}{número\ de\ respostas\ do\ padrão\ ouro\ para\ q} \quad (9)$$

$$precisão(q) = \frac{número\ de\ respostas\ corretas\ do\ sistema\ para\ q}{número\ de\ respostas\ do\ sistema\ para\ q} \quad (10)$$

A partir do cálculo da revocação e precisão, computaram-se as métricas F1-measure macro e micro. Para o cálculo da Escore F1 micro foram contados e somados todos os Verdadeiros Positivos, Falsos Positivos, Verdadeiros Negativos e Falsos Negativos obtidos para então calcular-se as micros precisão e revocação. Para o cálculo da Escore F1 macro foram calculadas a precisão, a revocação e o Escore F1 para cada questão e calculadas as respectivas médias.

Embora não seja uma comparação direta do desempenho obtido entre as abordagens devido ao uso de diferentes corpus de perguntas (ainda que as perguntas em português resultem da tradução das perguntas em inglês), dados os procedimentos seguidos na criação dos corpus usados para avaliação do sistema ENSEPRO (conforme apresentado na seção 6.4

Criação do corpus QALD-7 em Português Brasileiro), considera-se que a comparação dos desempenhos obtidos pelos sistemas é procedente.

Observa-se na Tabela 15 que o sistema ENSEPRO obteve os melhores resultados em quase todas as métricas, somente perdendo para o sistema ganswer2 no quesito de revocação micro. Atribui-se estes bons resultados principalmente ao uso extensivo das informações linguísticas em todo o decorrer do processo do sistema.

Conforme descrito no capítulo 4 Engenho semântico de Pergunta e Resposta orientado a Ontologias, o ENSEPRO é um sistema baseado em regras. Se por um lado sistemas baseados em regra tendem a ter boa precisão devido à quantidade de verdadeiros positivos, por outro podem apresentar revocação problemática devido ao fato de suas regras serem estáticas.

Pode-se compensar este aspecto dos sistemas baseados em regras pela implementação de um maior número de regras tentando-se prever todas as situações possíveis. Esta abordagem normalmente esbarra na dinamicidade do domínio em que o sistema está inserido. No caso de processamento de linguagem natural, a variação intrínseca somada a dinamicidade da linguagem dificulta consideravelmente a tentativa de prever-se todos os casos possíveis.

É com base neste contexto que se atribui o sucesso do sistema ENSEPRO ao uso das informações linguísticas da frase como base para suas regras de tomada de decisão. Ao fazer uso de níveis mais abstratos de informação linguísticas (níveis léxico, morfológico, sintático e estrutural), consegue-se compensar a falta de dinâmica do sistema de regras com o maior nível de generalização das regras pelo uso das informações linguísticas.

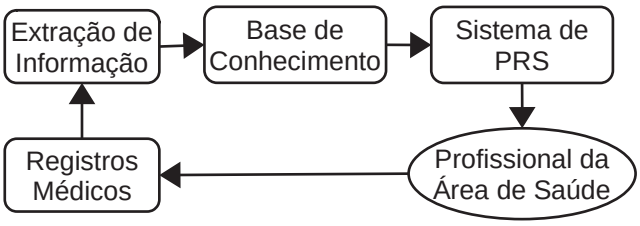
Os algoritmos de classificação do tipo de pergunta e seleção de palavras chaves (descritos respectivamente nas seções 4.2.3 Classificação do tipo de questão e 4.2.4 Seleção dos Termos Relevantes) corroboram com a hipótese descrita acima. Percebe-se que os algoritmos são bastante simples, com poucas regras, mas com resultados expressivos, como pode ser visto nas seções 6.1 Avaliação da classificação do tipo de questão e 6.3 Avaliação do processo de seleção de palavras chaves .

6.6 Avaliação de uso com ontologia na área da saúde

Tem-se como objetivo que o ENSEPRO seja um sistema de PRS independente de domínio. Foi com o intuito de comprovar estes objetivos que foram realizados os experimentos com a DBPedia em português. Mas, embora sejam duas BCs diferentes, o conhecimento representado nestas bases são semelhantes.

A fim de poder-se avaliar o quão genérico e independente de domínio é o sistema ENSEPRO, realizou-se um experimento de avaliação do sistema com uma BC da área da saúde, um contexto bastante diferente das avaliações realizadas anteriormente. Esta seção descreve a arquitetura geral utilizada neste estudo de caso, bem como os resultados obtidos. A Figura 15 apresenta uma visão geral dos principais componentes do projeto.

Figura 15: Visão geral dos componentes do contexto da BC.



Fonte: elaborada pelo autor.

O objetivo geral do projeto é apoiar as necessidades diárias dos profissionais de saúde, fornecendo um ecossistema de soluções visando melhorias na realização das suas atividades no acompanhamento de pacientes.

A BC utilizada no experimento de avaliação originou-se de um conjunto de 3.309 registros de profissionais da área de acompanhamento médico de pacientes ocorridos de abril de 2017 a novembro de 2018. Os dados armazenados na BC foram extraídos de um conjunto de documentos XML contendo 212.829 palavras. Estes registros contém tipicamente uma descrição do estado do paciente elaborada pelos médicos durante a realização de uma consulta médica.

A partir dos textos das anotações médicas são extraídas informações sobre o estado de saúde do paciente. Estas informações extraídas automaticamente são armazenadas de forma estruturada em uma BC. O processo de estruturação dos dados na BC possibilita o seu processamento por um sistema de PRS, o qual pode ser utilizado pelos profissionais da área de saúde para responder questões sobre o andamento do tratamento e o estado dos pacientes.

A partir do envolvimento de profissionais que utilizam um sistema médico para oncologia, foi possível definir um conjunto típico de sentenças de interesse. Essas sentenças estão descritas na Quadro 52 e constituem o primeiro passo do trabalho desenvolvido.

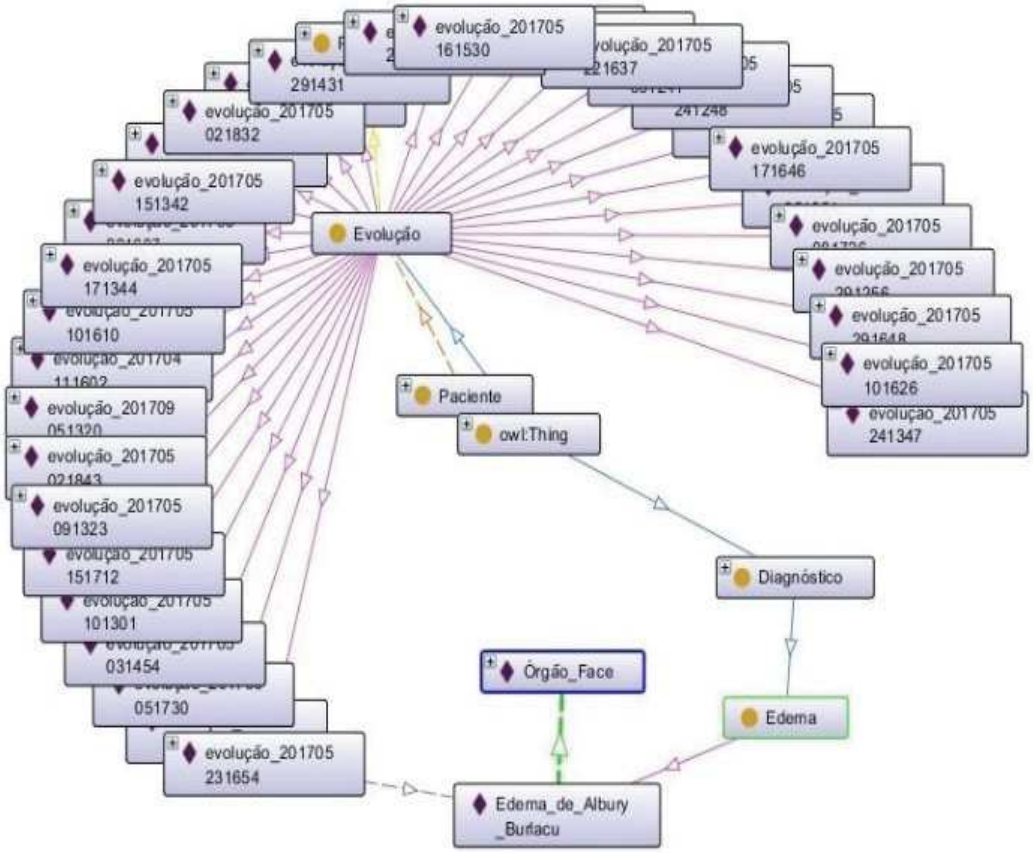
1. Quais os sintomas que o paciente {PacienteNome} relata?
2. Qual o diagnóstico do paciente {PacienteNome}?
3. Quais pacientes possuem o diagnóstico {Diagnostico}?
4. Quais sintomas os pacientes com diagnóstico {Diagnostico} relatam?
5. O sintoma {sintoma} está associado com quais diagnósticos?
6. Quais sintomas aparecem no órgão {orgao}?
7. Há quanto tempo paciente notou {sintoma} no órgão {orgao}?
8. Quais são os protocolos solicitados para o paciente {PacienteNome}?
9. Quais os medicamentos que o paciente {PacienteNome} faz uso?
10. Quais pacientes utilizam o medicamento {medicamento}?

Quadro 52: Conjunto de sentenças usados pelos profissionais de saúde.

A BC foi criada como uma ontologia no formato RDF. A ontologia foi projetada para representar as entidades e relações específicas identificadas nas sentenças selecionadas. Essa ontologia é usada pelo sistema ENSEPRO, que usa os nomes e relações das entidades para construir a consulta interna e a sentença de resposta. A ontologia utilizada no experimento é

composta por 53 classes e 12 relações. A Figura 16 mostra as principais entidades e relações da ontologia.

Figura 16: As principais classes da ontologia e suas respectivas instâncias.



Fonte: elaborada pelo autor.

O procedimento seguido para a criação da ontologia pode ser resumido aos seguintes passos: gerou-se um conjunto de documentos contendo textos com registros médicos reais. Esses dados foram exportados e anonimizados para preservar a privacidade de pacientes e profissionais. A partir deste ponto, realizaram entrevistas com profissionais de saúde envolvidos com o uso do sistema. Essas entrevistas tiveram como objetivo identificar o conjunto crítico de questões relevantes e fornecer os meios para modelar a BC. Com os principais conceitos e relações definidos, foi realizada uma análise dos textos para popular a BC. Após isto, submeteu-se ao sistema ENSEPRO a lista de perguntas utilizando-se a ontologia como BC.

Após a submissão das perguntas ao ENSEPRO, avaliou-se este estudo de caso com base em um modelo de cenário. Neste modelo de avaliação um avaliador descreve sistematicamente os cenários de interação do usuário com um sistema e avalia as ações necessárias para concluir cada um dos cenários executados pelo usuário com um modelo cognitivo (SUGIMURA; ISHIGAKI, 2005). Devido aos objetivos descritos em termos de cenários, o avaliador pode efetivamente identificar questões críticas que podem prejudicar o alcance dos objetivos.

Nesta avaliação de cenário, as situações práticas previstas pelo modelo proposto são previamente planejadas e descritas, antes de sua efetiva execução no sistema. Em seguida, comenta-se as etapas de sua execução e destaca-se os pontos relevantes para a avaliação do protótipo quanto à satisfação dos objetivos propostos. O cenário principal adotado visa demonstrar a correta identificação das informações necessárias. Portanto, consideramos como cenário de linha de base a situação em que a informação referida pelas sentenças de linguagem natural é corretamente identificada pela execução do sistema ENSEPRO sobre a base de conhecimento.

O Quadro 53 apresenta todas as informações geradas no processo de resposta a perguntas implementadas. Ele começa mostrando a sentença original em linguagem natural e, em seguida, mostra as respostas recuperadas da base de conhecimento. Cada resposta identifica a pontuação interna gerada e as classes e relações de ontologia identificadas. A consulta usada para acessar as respostas na base de conhecimento também é descrita. No final, as tuplas corretas da base de conhecimento são indicadas.

Quadro 53: Exemplo do retorno dado pelo sistema ENSEPRO a uma pergunta.

1. Questão:

"Quais os Sintomas que o paciente Marie Syamala relata?"

2. Respostas:

0 17.026 [distensao_abdominal | subclassof | *sintoma | *distensao_de_abdominal_marie_syamala_1 | type | distensao_abdominal] - [15.693 0.333 1.000]
1 14.575 [parada_de_eliminacao_de_gases | subclassof | *sintoma | *parada_de_eliminacao_de_gases_de_marie_syamala_1 | type | parada_de_eliminacao_de_gases] - [13.242 0.333 1.000]

3. Consulta SPARQL:

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX mas: <http://www..../marco#>
SELECT ?sintoma {mas:Marie_Syamala mas:possui ?evolucao. ?evolind mas:relata ? sintoma .}

4. Retorno da consulta:

mas:Distensão_de_Abdominal_Marie_Syamala_1
mas:Parada_de_Eliminação_de_Gases_de_Marie_Syamala_1

Neste caso, a geração de linguagem natural de uma sentença com a resposta não é executada, porque o foco deste estudo de caso não foi dedicado a esta etapa final do processo. O principal motivo para essa avaliação de processo foi obter indicações sobre o resultado positivo e perspectivas de uma automatização total do processo global. Os resultados de uma extensa análise experimental sobre um conjunto de sentenças previamente definidas (na Figura 4) e suas variações indicaram que a abordagem aplicada à extração de palavras-chave de sentenças originais foi eficaz em fornecer as informações necessárias para a descrição das consultas. O processo geral de análise manual dos textos médicos e a geração das instâncias da base de conhecimento forneceram os resultados positivos esperados. Estes resultados

foram computados considerando a anotação manual referente a cada sentença e as instâncias geradas pelos apontamentos obtidos no banco de dados do sistema de registros médicos. A análise de precisão das respostas recuperadas nas questões é de 87,5%, mesmo com um pequeno número de conceitos e relações presentes na base de conhecimento.

6.7 Análise dos resultados

O sucesso do sistema ENSEPRO depende da realização de uma sequência de processos para a compreensão da pergunta em um primeiro momento, seguido pela localização da resposta na BC. Todos os cinco processos realizados na etapa de compreensão da pergunta, por tratarem de processamento da linguagem natural, são passíveis de avaliação quanto ao desempenho. Se por um lado, analisando-se as opções disponíveis, decidiu-se pelo uso de implementações de terceiros para alguns processos (como o uso do parser Palavras para anotação linguística, o Spotlight como sistema de REM e a Wordnet para localização de sinônimos), por outro lado considerou-se mais adequado desenvolver um algoritmo próprio para os processos de classificação do tipo de pergunta e seleção de palavras-chave.

Por serem implementações próprias, considerou-se importante apresentar neste capítulo os resultados obtidos pelos algoritmos desenvolvidos. Pode-se observar que ambas as implementações são baseadas em regras e fazem uso das informações geradas pelo parser linguístico para realizarem suas atividades. Pode-se concluir pelos bons resultados obtidos em ambos os experimentos de avaliação (Escore F1 médio de 0,9769 na classificação de tipo de pergunta e 0,9868 na seleção de palavras chaves) que o uso das informações linguísticas possibilita a implementação de processos com excelentes resultados.

Os resultados apresentados corroboram a hipótese de que as características linguísticas de frases fornecem uma ampla fonte de subsídios para a implementação de algoritmos para o processamento da linguagem natural. Quanto mais alto o nível de abstração linguística, tanto melhor o desempenho do processo. Este aspecto é demonstrado pelo bom desempenho apresentado pelo sistema ENSEPRO no principal experimento de avaliação deste trabalho, com o escore F1, o qual avalia a capacidade de localização das respostas para as perguntas do QALD-7. Estes resultados reforçam a hipótese de que as informações linguísticas têm subsídios importantes para a compreensão da linguagem natural e podem colaborar significativamente no desempenho de sistemas de PRS.

De uma forma geral, pode-se observar que os algoritmos que fazem uso das informações linguísticas para realizar a compreensão da linguagem natural são, via de regra, pouco complexos. Analisando-se os algoritmos do classificador de tipo de questões e do seletor de palavras-chaves, verifica-se que com poucas regras é possível implementar-se processos com alto grau de generalidade e com excelentes desempenhos.

Os algoritmos do próprio sistema ENSEPRO, os quais fazem uso das informações linguísticas em diversos pontos do processo, aproveitam-se das informações linguísticas para refinar suas decisões e evitar com que triplas contendo informações irrelevantes façam parte do processo de seleção de respostas. Considerou-se que a hipótese de que sistemas de PRS possam obter melhores resultados a partir do uso combinado de diferentes técnicas (linguística, grafos e AM, por exemplo) confirmou-se nos resultados do sistema ENSEPRO em relação ao QALD-7.

Outra questão importante a ser comentada é o experimento de uso do sistema ENSEPRO para responder perguntas relacionadas a informações representadas em uma ontologia da área da saúde. Ainda que estes experimentos tenham sido realizados de forma preliminar, estes indicam que o uso combinado de diferentes técnicas, tomando como base as informações linguísticas das perguntas, possibilitam um nível de compreensão da linguagem natural de forma independente do domínio da ontologia. É importante salientar novamente que esta é uma hipótese ainda a ser comprovada, mas os experimentos preliminares indicam boas perspectivas.

Por fim, é importante fazer alguns comentários em relação ao processo de criação do corpus em português para avaliação do ENSEPRO. O tamanho do corpus (215 perguntas no total) não pode ser considerado como extenso em relação aos padrões atuais, principalmente em se tratando de algoritmos baseados em Aprendizagem de Máquina. Conforme comentado no capítulo de introdução desta tese, percebe-se uma busca bastante intensa de soluções baseadas em Aprendizagem de Máquina, visando principalmente a independência de idioma.

O esforço empreendido para a geração de uma versão em português do QALD-7, bem como a sua adaptação ao conteúdo da DBPedia em português, evidenciaram de forma veemente o quão complexo e trabalhoso é produzir tais artefatos. Para línguas com quantidade reduzida de corpus anotados, como é o caso da língua portuguesa, a escassez de recursos semanticamente anotados faz com que na prática restrinjam-se as abordagens de processamento da linguagem natural às técnicas de Aprendizagem de Máquina não supervisionadas ou sistemas baseados em regras.

Em relação às três principais técnicas para implementação de sistemas de PRS (linguística, grafo e AM, sendo as duas primeiras técnicas baseadas em regras), somente as abordagens baseadas em informações linguísticas fazem uso de informações dos níveis mais altos de abstração da linguagem. As técnicas baseadas em grafos e AM não supervisionada utilizam somente as informações léxicas e morfológicas da linguagem para extrair significado. Defende-se nesta tese que o uso combinado das técnicas não só é possível, como incrementa positivamente a precisão e revocação do sistema. Isso é observado nos resultados obtidos pelos Escore F1 macro com valor médio de 11,46% superior em relação aos observados nos trabalhos dos demais participantes do QALD-7.

Assim sendo, segundo o que se observou no decorrer deste trabalho, considerando-se principalmente os resultados obtidos nos experimentos de avaliação, ao estipular-se como objetivo o desenvolvimento de um sistema de PRS para o processamento de perguntas curtas em português, dado o fato de não ser uma língua que conte com grande disponibilidade de corpus anotados, entende-se ser de grande valia a opção por abordagens que apoiem-se nas informações de nível mais elevado da linguagem, principalmente aquelas que buscam o uso combinado com outras técnicas.

7 CONSIDERAÇÕES FINAIS

O trabalho desenvolvido nesta tese teve como principal objetivo verificar a hipótese de que um sistema com abordagem híbrida incorporando informações linguísticas de perguntas pode otimizar o desempenho um sistema de PRS independente de domínio.

Para alcançar este objetivo, realizou-se um levantamento do estado da arte em relação às abordagens atualmente utilizadas para a implementação dos sistemas de PRS. A partir deste levantamento, projetou-se um modelo para o desenvolvimento de um sistema de PRS para responder perguntas curtas expressas em português.

O principal diferencial da abordagem desenvolvida neste trabalho é a exploração aprofundada das informações linguísticas da pergunta primeiramente para a compreensão da pergunta e, posteriormente, para a otimização do desempenho da consulta, geração, ranqueamento e seleção de respostas, processos estes baseados em busca em grafos. Além da combinação das técnicas baseadas em informações linguísticas e busca em grafos, considera-se também um diferencial o uso de word embedding para a implementação de um algoritmo complementar de seleção e validação de respostas.

Uma vez definido o modelo, implementou-se o protótipo do sistema, o qual recebeu o nome de ENSEPRO, um acrônimo para Engenho Semântico de Pergunta e Resposta para Ontologias. A partir deste protótipo realizou-se quatro experimentos de avaliação.

O dois primeiros experimentos de avaliação visaram a verificação do desempenho de dois algoritmos do fluxo de processos do módulo de Compreensão da Linguagem Natural. O terceiro experimento avaliou a capacidade de localização de respostas corretas do sistema ENSEPRO, sendo o quarto experimento desenvolvido para avaliar a aplicação da abordagem sobre uma ontologia da área da saúde.

Os desempenhos obtidos nestes experimentos de avaliação foram registrados e apresentados. Conforme visto na análise dos resultados, considera-se que a melhoria média de 16,04% na precisão macro e 5,7% na revocação macro (resultando em um Escore F1 macro 11,46% maior na média) valida a hipótese de que as informações linguísticas dos níveis de abstração mais elevados contribuem para a compreensão da linguagem natural, otimizando o desempenho geral de sistemas de PRS.

Da caminhada do trabalho de pesquisa descrito acima resultou um conjunto de contribuições científicas, as quais são elencadas na próxima seção.

7.1 Contribuições científicas

Com relação ao legado científico desta tese de doutorado, apresenta-se como contribuição a disponibilização de um modelo para implementação de um sistema de PRS que possibilita a exploração de forma mais aprofundada das informações linguísticas de níveis mais altos de abstração para a implementação de algoritmos mais precisos de compreensão da LN.

Geralmente os sistemas de PRS limitam-se a explorar os níveis léxico e morfológicos das palavras, restringindo-se assim a compreensão do significado das palavras aos dois níveis

linguísticos mais elementares. O modelo apresentado nesta tese demonstra explicitamente o uso das informações sintáticas e estruturais das frases em LN para a implementação de algoritmos que visam a compreensão da pergunta.

Considera-se também como uma contribuição a implementação do modelo proposto em um protótipo e a sua respectiva disponibilização como um projeto de código aberto (open source) em repositório público bem conhecido[44]. Agindo desta forma tem-se em vista, além da devida transparência em relação ao trabalho desenvolvido, compartilhar-se os conhecimentos adquiridos, visando não somente a colaboração com o projeto, mas principalmente servindo como um estímulo para a continuidade da pesquisa em trabalhos relacionados à área do PLN e mais especificamente de sistemas de PRS.

Considera-se também como outra contribuição importante desta tese de doutorado o objetivo de disponibilizar um modelo e um protótipo de sistema de PRS com desempenho satisfatório específico para a língua portuguesa. Conforme comentado no capítulo de introdução, percebeu-se que há uma quantidade significativa de trabalhos que propõem abordagens independentes de idioma, mas poucos são aqueles que apresentam resultados para a língua portuguesa.

Além do modelo e do protótipo do sistema de PRS, é também legado da pesquisa desenvolvida nesta tese de doutorado um conjunto de publicações em periódicos científicos e apresentações em eventos relacionados à área da computação. No decorrer do doutorado foram publicados os seguintes trabalhos:

- Artigos completos em periódicos
  1. DE ARAUJO, DENIS ANDREI; RIGO, SANDRO JOSÉ; BARBOSA, JORGE LUIS VICTÓRIA. Ontology-based information extraction for juridical events with case studies in Brazilian legal realm. ARTIFICIAL INTELLIGENCE AND LAW (DORDRECHT. PRINT), v.25, p. 1-18, 2017. Referências adicionais : Inglês. Meio de divulgação: Meio digital. Home page: [doi:10.1007/s10506-017-9203-z]
- Artigos aceitos para publicação
  2. ARAUJO, D. A.; HENTGES, A.; RIGO, SANDRO J.; RIGHI, RODRIGO DA ROSA Applying parallelization strategies for inference mechanisms performance improvement. IEEE Latin America Transactions, 2018.
  3. BERLITZ, E. E.; ARAUJO, D. A.; Silva, Allan de Barcelos; RIGHI, R. R.; RIGO, SANDRO J. Distributional Models with Syntactic Contexts for the Measurement of Word Similarity in Brazilian Portuguese. JOURNAL OF COMPUTER SCIENCES, 2019.
- Trabalhos publicados em anais de eventos (completo)
  4. ARAUJO, D. A.; SCHWERTZNER, M. A.; rigo, SANDRO J.; SKOFIER, B.; SILVA, A. B. Fostering natural language question answering over knowledge bases in oncology EHR In: CBMS - International Symposium on Computer-Based Medical Systems, 2019, Corboba. CBMS - International Symposium on Computer-Based Medical Systems. córdoba: IEEE CBMS, 2019. v.1. p.1 - 10.
  5. RODRIGUES, E. L. F.; SILVA, A. L.; ARAUJO, D. A.; RIGO, SANDRO J. Natural language question generation from Connected Open Data: a study of possibilities In: XLIV Conferência Latino-americana de Informática, 2018, São

[44] https://github.com/Ensepro

Paulo. XLIV Conferência Latino-americana de Informática. São Paulo: CLEI, 2018. v.1. p.1 - 10

9. ARAUJO, D. A.; RIGO, S. J.; HENTGES, A. Uma abordagem linguística para sistemas de Perguntas e Respostas Curtas In: Simpósio Brasileiro de Sistemas de Informação, 2018, Caxias do Sul/RS.
10. ARAUJO, D. A.; RIGO, S. J.; DAMBROS, V.; MARTINI, A.; DAUBER, D.; SANTOS, R. P. Qualis SR: Um Sistema Híbrido de Recomendação de Tratamento para Infecções In: Escola Regional de Computação Aplicada à Saúde (ERCAS 2016), 2016, Novo Hamburgo. Escola Regional de Computação Aplicada à Saúde (ERCAS 2016), 2016.

- Trabalhos publicados em anais de eventos (resumo expandido)

11. ARAUJO, D. A.; RIGO, S. J.; RIGHI, R. Paralelização do Algoritmo RETE usando Threads In: XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2016), 2016, São Leopoldo/RS. XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2016). Porto Alegre: Sociedade Brasileira de Computação, 2016. v.1. p.191 - 192

- Palestras em eventos

12. ARAUJO, D. A.; SCHWERTZNER, M. A.; rigo, SANDRO J.; SKOFIER, B. Fostering natural language question answering over knowledge bases in oncology EHR In: Inteligência Artificial e Sistemas Inteligentes na Saúde: Oportunidades e Desafios. Porto Alegre: UFCSPA, 2019.

Pode-se apontar também como uma contribuição o estágio de pesquisa realizado na Universidade de Évora em Portugal entre agosto de 2018 e fevereiro de 2019, pois além da troca de experiências com o grupo de pesquisa do Prof. Dr. Paulo Quaresma, possibilitou a participação em eventos acadêmicos, como a oficina “Practical use of linked data”, reuniões de apresentação das pesquisas, bem como a troca de experiências com os doutorandos do grupo de pesquisa.

São também contribuições a disponibilização de dois corpus para avaliação de sistemas de PRS que visem especificamente o processamento de perguntas formuladas em português. O primeiro corpus refere-se a tradução de 268 questões do conjunto de dados para avaliação de sistemas de PRS do evento QALD-7. Este trabalho de tradução das questões para o português falado no Brasil foi oficialmente integrado e disponibilizado diretamente repositório público do evento[45]. Uma derivação deste corpus foi desenvolvida, visando possibilitar o uso da versão em português da DBPedia (o corpus do QALD visa o uso da DBPedia em inglês), sendo este novo corpus também disponibilizado publicamente[46].

Por fim, aponta-se como contribuição científica as possibilidades de continuidade da pesquisa em trabalhos futuros, assunto da próxima seção.

[45] https://github.com/ag-sc/QALD/tree/master/7
[46] https://github.com/DenisAraujo68/ensepro-qald-7_pt-br

7.2 Trabalhos Futuros

Realizados os experimentos, confirmada a viabilidade da proposta e verificado o desempenho pretendido, despontam possibilidades de continuidade deste trabalho, seja do ponto de vista do aprofundamento do potencial da abordagem através de novos experimentos, seja pela investigação de novos campos de pesquisa para a sua aplicação.

Em relação ao protótipo desenvolvido, ressente-se a falta de uma camada de abstração para as ferramentas externas, como o parser linguístico e o sistema REM. A abstração das ferramentas externas traria como resultado uma maior independência da abordagem proposta. Ainda em relação à independência, seria muito interessante a utilização de padrões de anotação multilinguísticos, como o Universal Dependency (MCDONALD et al., 2013).

Outra questão importante em relação à implementação proposta diz respeito ao algoritmo de geração de triplas candidatas à resposta. A implementação faz uso massivo de paralelização e, por isso, tem um tempo de execução aceitável. Contudo, sabe-se haver alternativas disponíveis que, em teoria, possibilitariam reduzir significativamente o tempo necessário para geração de combinação de triplas candidatas. O uso de representações binárias do RDF como o HDT (FERNÁNDEZ et al., 2013) poderia reduzir o tempo necessário para combinação das triplas candidatas, possibilitando extrapolar-se a atual limitação de geração de pares de triplas candidatas, permitindo a geração de combinações mais longas, como trincas, quadras ou quintetos de triplas.

Ainda em relação às triplas candidatas, vê-se oportunidade de trabalhos futuros em relação ao uso de Aprendizagem de Máquina para o ranqueamento e seleção de candidatas. Como pode ser visto na seção 4.3.4 Ranqueamento de respostas, o algoritmo de ranqueamento baseia-se nas características morfossintáticas dos elementos da tripla em relação aos Termos Relevantes da pergunta. Cada vez que um pergunta é submetida ao sistema implementado, são geradas centenas de milhares a dezenas de milhões de triplas candidatas, as quais são individualmente analisadas.

Intuitivamente, supõe-se que o algoritmo de ranqueamento é previsível o suficiente para ser aprendido por alguma técnica de Aprendizagem de Máquina com algoritmo mais eficiente em termos de tempo de execução. O algoritmo de ranqueamento atual poderia facilmente gerar bilhões de exemplos para treinamento supervisionado do algoritmo de AM.

Em relação ao algoritmo de seleção das melhores candidatas, atualmente este é realizado pela ordenação da lista de triplas candidatas utilizando o valor do ranqueamento como chave. O algoritmo de ordenação é intrinsecamente sequencial, diminuindo significativamente o desempenho do sistema. Novamente, acredita-se que seria possível utilizar um classificador baseado em AM para a seleção ou uma triagem preliminar das candidatas. O uso de um classificador possibilitaria a análise paralela das candidatas, possibilitando um incremento substancial no tempo de execução do processo de seleção.

Vislumbra-se ainda uma última melhoria operacional em relação ao protótipo atualmente implementado: o modelo de word embedding a ser utilizado no processo de seleção e validação de respostas. Percebeu-se no decorrer dos experimentos de avaliação do protótipo que o tipo e o modelo de word embedding utilizado é um fator crucial para o funcionamento da seleção e validação das respostas. Devido ao tempo disponível, não foi

possível realizar-se uma bateria de testes mais abrangente para verificação de qual modelo de word embedding proporcionaria melhor desempenho.

Em relação ao nível de independência de domínio do modelo proposto, pretende-se realizar novos experimentos do protótipo na área da saúde. Conforme relatado na seção 6.6 Avaliação de uso com ontologia na área da saúde, foi realizado um experimento para verificar-se preliminarmente a viabilidade da aplicação do protótipo no contexto de um trabalho de mestrado em andamento. Tem-se a intenção de realizar novas avaliações tão logo esteja disponível a nova versão da ontologia.

Aliás, é também trabalho futuro a integração do protótipo implementado com outros trabalhos do grupo de pesquisa. Pretende-se, por exemplo, realizar experimentos de avaliação do uso do modelo DEPS de word embedding (LEVY; GOLDBERG, 2014) no algoritmo de seleção e validação de respostas da abordagem aqui proposta.

No que tange ainda à integração com outras pesquisas, percebe-se que o escopo do modelo apresentado nesta tese abrange os processos de compreensão da pergunta e busca e seleção das respostas na BC. Contudo, um sistema de PRS tem também como objetivo a geração da resposta em LN. O protótipo foi implementado tendo em vista a sua integração com outros trabalhos sobre a Geração de Linguagem Natural do grupo de pesquisa do Prof. Dr. Sandro Rigo, orientador desta tese de doutorado.

REFERÊNCIAS

ABDUL-KADER, S. A.; WOODS, J. C. Survey on chatbot design techniques in speech conversation systems. **International Journal of Advanced Computer Science and Applications**, v. 6, n. 7, 2015.

ALVES, I. M. Neologia e níveis de análise lingüística. **As Ciências do léxico**, p. 77–91, 2001.

BAUDIŠ, P. **YodaQA: a modular question answering system pipeline**. POSTER 2015-19th International Student Conference on Electrical Engineering. **Anais**...2015

BECHARA, E. **Moderna gramática portuguesa - Revista, ampliada e atualizada conforme o novo Acordo Ortográfico**. 37. ed. [s.l.] Nova Fronteira, 2012.

BENVENISTE, É. Os níveis de análise linguística. **E. BENVENISTE, Problemas de Linguística Geral I. Campinas, Pontes**, p. 127–140, 1964.

BERANT, J. et al. **Semantic parsing on freebase from question-answer pairs**. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. **Anais**...2013

BERNERS-LEE, T. et al. The semantic web. **Scientific american**, v. 284, n. 5, p. 28–37, 2001.

BICK, E. **The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework**. [s.l.] Aarhus Universitetsforlag, 2000.

BLAIR, D. C. An evaluation of retrieval effectiveness for a full-text document retrieval system. 1984.

BOBROW, D. G. Natural language input for a computer problem solving system. 1964.

BORDES, A. et al. Large-scale simple question answering with memory networks. **arXiv preprint arXiv:1506.02075**, 2015.

BORST, W. N. Construction of engineering ontologies. **University of Twenty, Enschede, Center for Telematica and Information Technology**, 1997.

BURON, M. et al. **Reformulation-based query answering for RDF graphs with RDFS ontologies**. ESWC 2019. **Anais**...2019

CHEN, Y.; ZHANG, J.; ZHANG, C. **Automatic natural language processing based data extraction**, fev. 2019.

CIMIANO, P. et al. **Multilingual question answering over linked data (qald-3): Lab overview**. International Conference of the Cross-Language Evaluation Forum for European Languages. **Anais**...Springer, 2013

COHEN, W.; RAVIKUMAR, P.; FIENBERG, S. **A comparison of string metrics for matching names and records**. Kdd workshop on data cleaning and object consolidation. **Anais**...2003Disponível em: <https://www.cs.cmu.edu/afs/cs/Web/People/wcohen/postscript/kdd-2003-match-ws.pdf>. Acesso em: 23 abr. 2017

COVINGTON, M. A. **Natural language processing for Prolog programmers**. [s.l.] Prentice hall Englewood Cliffs (NJ), 1994.

CUNNINGHAM, H. et al. **GATE: an architecture for development of robust HLT applications**. . In: PROCEEDINGS OF THE 40TH ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. Association for Computational Linguistics, 7 jun. 2002Disponível em: <http://dl.acm.org/citation.cfm?id=1073083.1073112>. Acesso em: 22 abr. 2017

DAMLJANOVIC, D.; AGATONOVIC, M.; CUNNINGHAM, H. **Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction**. Extended Semantic Web Conference. **Anais**...Springer,

2010Disponível em: <http://link.springer.com/10.1007%2F978-3-642-13486-9_8>. Acesso em: 5 set. 2016
DAMLJANOVIC, D.; AGATONOVIC, M.; CUNNINGHAM, H. **FREyA: An interactive way of querying Linked Data using natural language**. Extended Semantic Web Conference. **Anais**...Springer, 2011
DE ARAUJO, D. A.; HENTGES, A. R.; RIGO, S. **A Linguistic Approach to Short Query and Answer Systems**. Proceedings of the XIV Brazilian Symposium on Information Systems. **Anais**...ACM, 2018
DE PAIVA, V. et al. **NomLex-PT: A Lexicon of Portuguese Nominalizations.** LREC. **Anais**...2014
DECKER, S. et al. The semantic web: The roles of XML and RDF. **IEEE Internet computing**, v. 4, n. 5, p. 63–73, 2000.
DEGEN, W. et al. **GOL: A general ontological language**. Formal Ontology and Information Systems. **Anais**...Citeseer, 2001Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.818>. Acesso em: 1 jul. 2017
DIEFENBACH, D. et al. Core techniques of question answering systems over knowledge bases: a survey. **Knowledge and Information systems**, v. 55, n. 3, p. 529–569, 2018a.
DIEFENBACH, D. et al. Towards a Question Answering System over the Semantic Web. **arXiv preprint arXiv:1803.00832**, 2018b.
DIEFENBACH, D.; SINGH, K.; MARET, P. **WDAqua-core1: A Question Answering service for RDF Knowledge Bases**. Companion of the The Web Conference 2018 on The Web Conference 2018. **Anais**...International World Wide Web Conferences Steering Committee, 2018
FENSEL, D. et al. OIL: An ontology infrastructure for the semantic web. **IEEE intelligent systems**, v. 16, n. 2, p. 38–45, 2001.
FERNÁNDEZ, J. D. et al. Binary RDF representation for publication and exchange (HDT). **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 19, p. 22–41, 2013.
FUTRELL, R. L.; GRUBER, T. R. **Exemplar-based natural language processing**, jan. 2019.
GONDEK, D. C. et al. A framework for merging and ranking of answers in DeepQA. **IBM Journal of Research and Development**, v. 56, n. 3.4, p. 14–1, 2012.
GRUBER, T. R. **Ontolingua: A mechanism to support portable ontologies**. [s.l.] Stanford University, Knowledge Systems Laboratory Stanford, 1992.
GUARINO, N.; OBERLE, D.; STAAB, S. What is an Ontology? In: **Handbook on ontologies**. [s.l.] Springer, 2009. p. 1–17.
HALL, R. A. **External history of the Romance languages**. [s.l.] Elsevier Publishing Company, 1974. v. 2
HAMON, T.; GRABAR, N.; MOUGIN, F. Querying biomedical linked data with natural language questions. **Semantic Web**, v. 8, n. 4, p. 581–599, 2017.
HE, S. et al. CASIA@ V2: A MLN-based question answering system over linked data. 2014.
HEBERT, M. et al. **Multi-domain natural language processing architecture**, maio 2019.
HENDLER, J.; MCGUINNESS, D. L. The DARPA agent markup language. **IEEE Intelligent systems**, v. 15, n. 6, p. 67–73, 2000.
HERMJAKOB, U. **Parsing and question classification for question answering**. Proceedings of the ACL 2001 workshop on open-domain question answering. **Anais**...2001

HÖFFNER, K. et al. Survey on Challenges of Question Answering in the Semantic Web. **Submitted to the Semantic Web Journal**, 2016.
HÖFFNER, K. et al. Survey on challenges of question answering in the semantic web. **Semantic Web**, v. 8, n. 6, p. 895–920, 2017.
HOFWEBER, T. Logic and Ontology. In: ZALTA, E. N. (Ed.). . **The Stanford Encyclopedia of Philosophy**. Fall 2014 ed. [s.l.] Metaphysics Research Lab, Stanford University, 2014.
HOLMES, D.; MCCABE, M. C. **Improving precision and recall for soundex retrieval**. Information Technology: Coding and Computing, 2002. Proceedings. International Conference on. **Anais**...IEEE, 2002Disponível em: <http://ieeexplore.ieee.org/abstract/document/1000354/>. Acesso em: 23 abr. 2017
HORROCKS, I.; OTHERS. DAML+OIL: A Description Logic for the Semantic Web. **IEEE Data Eng. Bull.**, v. 25, n. 1, p. 4–9, 2002.
HOSSIN, M.; SULAIMAN, M. N. A review on evaluation metrics for data classification evaluations. **International Journal of Data Mining & Knowledge Management Process**, v. 5, n. 2, p. 1, 2015.
HU, X. et al. Natural language aggregate query over RDF data. **Information Sciences**, v. 454, p. 363–381, 2018.
HUANG, R.; ZOU, L. **Natural language Question Answering over RDF data**. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. **Anais**...ACM, 2013
JACQUEMIN, C.; TZOUKERMANN, E. NLP for term variant extraction: synergy between morphology, lexicon, and syntax. In: **Natural language information retrieval**. [s.l.] Springer, 1999. p. 25–74.
KETSMUR, M.; RODRIGUES, M.; TEIXEIRA, A. DBPEDIA BASED FACTUAL QUESTIONS ANSWERING SYSTEM. **IADIS International Journal on WWW/Internet**, v. 15, n. 1, 2017.
KLYNE, G.; CARROLL, J. J. Resource description framework (RDF): Concepts and abstract syntax. 2006.
KNÖPFEL, A. **FMC Quick Introduction.** [s.l.] FMC Consortium, 2007.
KONONENKO, O. et al. **Mining modern repositories with elasticsearch**. Proceedings of the 11th Working Conference on Mining Software Repositories. **Anais**...ACM, 2014
LEVENSHTEIN, V. I. **Binary codes capable of correcting deletions, insertions, and reversals**. Soviet physics doklady. **Anais**...1966
LEVY, A.; RAJARAMAN, A.; ORDILLE, J. Query answering algorithms for information agents. 1996.
LEVY, O.; GOLDBERG, Y. **Dependency-based word embeddings**. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). **Anais**...2014
LIDDY, E. D. et al. Natural language processing. **Encyclopedia of library and information science**, v. 2, 2003.
MANNING, C. D.; MANNING, C. D.; SCHÜTZE, H. **Foundations of statistical natural language processing**. [s.l.] MIT press, 1999.
MCBRIDE, B. The resource description framework (RDF) and its vocabulary description language RDFS. In: **Handbook on ontologies**. [s.l.] Springer, 2004. p. 51–65.
MCDONALD, R. et al. **Universal dependency annotation for multilingual parsing**. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). **Anais**...2013

MCGUINNESS, D. L.; VAN HARMELEN, F.; OTHERS. OWL web ontology language overview. **W3C recommendation**, v. 10, n. 10, p. 2004, 2004.
METZLER, D.; CROFT, W. B. Analysis of statistical question classification for fact-based questions. **Information Retrieval**, v. 8, n. 3, p. 481–504, 2005.
MOLLÁ, D.; VICEDO, J. L. Question answering in restricted domains: An overview. **Computational Linguistics**, v. 33, n. 1, p. 41–61, 2007.
MURDOCK, J. W. et al. Typing candidate answers using type coercion. **IBM Journal of Research and Development**, v. 56, n. 3.4, p. 7–1, 2012.
NGOMO, N. 9th Challenge on Question Answering over Linked Data (QALD-9). **language**, v. 7, p. 1, 2019.
PAŞCA, M. **Open-domain question answering from large text collections**. [s.l.] MIT Press, 2003.
POSNER, R. **The romance languages**. [s.l.] Cambridge University Press, 1996.
PRODANOV, C. C.; DE FREITAS, E. C. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico-2ª Edição**. [s.l.] Editora Feevale, 2013.
PRUD, E.; SEABORNE, A. SPARQL query language for RDF. 2006.
RADFORD, A. et al. Improving language understanding by generative pre-training. **URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf**, 2018.
RAMOS, J. **Using tf-idf to determine word relevance in document queries**. Proceedings of the first instructional conference on machine learning. **Anais**...2003
RIETVELD, L.; HOEKSTRA, R. The YASGUI family of SPARQL clients 1. **Semantic Web**, v. 8, n. 3, p. 373–383, 2017.
SAWANT, U.; CHAKRABARTI, S.; RAMAKRISHNAN, G. Open-domain Question Answering Using a Knowledge Graph and Web Corpus. **SIGWEB Newsl.**, n. Winter, p. 4:1–4:8, mar. 2018.
SHEKARPOUR, S. et al. Sina: Semantic interpretation of user queries for question answering on interlinked data. **Journal of Web Semantics**, v. 30, p. 39–51, 2015.
SILVA, J. et al. From symbolic to sub-symbolic information in question classification. **Artificial Intelligence Review**, v. 35, n. 2, p. 137–154, 2011.
STAMP, M. A revealing introduction to hidden Markov models. **Department of Computer Science San Jose State University**, p. 26–56, 2004.
STRZALKOWSKI, T.; HARABAGIU, S. **Advances in open domain question answering**. [s.l.] Springer Science & Business Media, 2006. v. 32
STUPINA, A. A. et al. **Question-answering system**. IOP Conference Series: Materials Science and Engineering. **Anais**...IOP Publishing, 2016Disponível em: <http://iopscience.iop.org/article/10.1088/1757-899X/155/1/012024/meta>. Acesso em: 24 jul. 2017
SUGIMURA, V. S. S. V. M.; ISHIGAKI, V. K. New web-usability evaluation method: scenario-based walkthrough. **FUJITSU Sci. Tech. J**, v. 41, n. 1, p. 105–114, 2005.
SUN, J.; JIN, Q. **Scalable rdf store based on hbase and mapreduce**. 2010 3rd international conference on advanced computer theory and engineering (ICACTE). **Anais**...IEEE, 2010
SWANSON, D. R. Searching natural language text by computer. **Science**, v. 132, n. 3434, p. 1099–1104, 1960.
UNGER, C. et al. **Question answering over linked data (QALD-4)**. Working Notes for CLEF 2014 Conference. **Anais**...2014

USBECK, R. et al. **7th open challenge on question answering over linked data - QALD7**. Semantic Web Evaluation Challenge. **Anais**...Springer, 2017
USBECK, R. et al. 8th Challenge on Question Answering over Linked Data (QALD-8). **language**, v. 7, p. 1, 2018.
VIEIRA, R.; LIMA, V. L. **Lingüística computacional: princípios e aplicações**. Anais do XXI Congresso da SBC. I Jornada de Atualização em Inteligência Artificial. **Anais**...sn, 2001Disponível em: <http://www.inf.unioeste.br/~jorge/MESTRADOS/LETRAS%20-%20MECANISMOS%20DO%20FUNCIONAMENTO%20DA%20LINGUAGEM%20-%20PROCESSAMENTO%20DA%20LINGUAGEM%20NATURAL/ARTIGOS%20INTERESSANTES/lingu%EDstica%20computacional.pdf>. Acesso em: 25 mar. 2017
VINYALS, O.; LE, Q. A neural conversational model. **arXiv preprint arXiv:1506.05869**, 2015.
VOIGT, M.; MITSCHICK, A.; SCHULZ, J. **Yet Another Triple Store Benchmark? Practical Experiences with Real-World Data.** SDA. **Anais**...Citeseer, 2012
WANG, R.; LING, Z.; HU, Y. Knowledge Base Question Answering With Attentive Pooling for Question Representation. **IEEE Access**, v. 7, p. 46773–46784, 2019.
WANTROBA, E. J.; ROMERO, R. A. F. **A method for designing dialogue systems by using ontologies**. IEEE/RSJ International Conference on Intelligent Robots and Systems; Workshop on Standardized Knowledge Representation and Ontologies for Robotics and Automation, 1st. **Anais**...Institute of Electrical and Electronics Engineers-IEEE, 2014Disponível em: <http://www.producao.usp.br/handle/BDPI/48886>. Acesso em: 19 jul. 2016
WAZLAWICK, R. **Metodologia de pesquisa para ciência da computação**. [s.l.] Elsevier Brasil, 2015. v. 2
WAZLAWICK, R. **Metodologia de pesquisa para ciência da computação**. [s.l.] Elsevier Brasil, 2017. v. 2
WERMTER, S.; RILOFF, E.; SCHELER, G. **Connectionist, statistical and symbolic approaches to learning for natural language processing**. [s.l.] Springer Science & Business Media, 1996. v. 1040
XU, K.; FENG, Y.; ZHAO, D. **Xser@ qald-4: Answering natural language questions via phrasal semantic parsing**. Working Notes for CLEF 2014 Conference. **Anais**...2014
XU, P.; SARIKAYA, R. **Contextual domain classification in spoken language understanding systems using recurrent neural network**. Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. **Anais**...IEEE, 2014Disponível em: <http://ieeexplore.ieee.org/abstract/document/6853573/>. Acesso em: 24 jul. 2017
YANG, H. et al. **Structured use of external knowledge for event-based open domain question answering**. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. **Anais**...ACM, 2003Disponível em: <http://dl.acm.org/citation.cfm?id=860444>. Acesso em: 24 jul. 2017
YOUNG, T. et al. Recent trends in deep learning based natural language processing. **ieee Computational intelligenCe magazine**, v. 13, n. 3, p. 55–75, 2018.
ZOU, L. et al. **Natural Language Question Answering over RDF: A Graph Data Driven Approach**. Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. **Anais**...: SIGMOD '14.New York, NY, USA: ACM, 2014Disponível em: <http://doi.acm.org/10.1145/2588555.2610525>. Acesso em: 21 maio. 2018

APÊNDICE A - OPÇÕES DO SISTEMA ENSEPRO

```
usage: ensepro [-h] [-frase FRASE] [-arquivo-frases ARQUIVO_FRASES]
                       [-vec-file WORD_EMBEDDING_VECTOR] [-bin-vec]
                       [-glove-vec] [-save-json] [-save-txt] [-tr] [-sin]
                       [-cn] [-lv] [-arvore] [-tags] [-resposta] [-verbose]
                       [-quiet] [-original] [-final] [--sem-resposta]
                       [-somente-resposta]

optional arguments:
  -h, --help            show this help message and exit
  -frase FRASE          Frase a ser analisada. (default: None)
  -arquivo-frases ARQUIVO_FRASES
                              Arquivo contento frases a serem analisadas.
(default:
                        None)
  -vec-file WORD_EMBEDDING_VECTOR
                        Word Embedding vector file. (default: None)
  -bin-vec              Indica que o arquivo vec é um arquivo binário.
                        (default: False)
   -glove-vec               Indica que o arquivo vec é um arquivo glove.
(default:
                        False)
  -save-json            Salvará os resultdos em um arquivo json. (default:
                        False)
  -save-txt             Salvará os resultdos em um arquivo txt. (default:
                        False)
  -tr                   Indica para printar/salvar os termos relevantes.
                        (default: False)
  -sin                  Indica para printar/salvar os sinonimos dos termos
                        relevantes. (default: False)
   -cn                         Indica para printar/salvar os complementos
nominais.
                        (default: False)
  -lv                   Indica para printar/salvar os locucoes verbais.
                        (default: False)
   -arvore                  Indica para printar/salvar a arvore gráfica da
frase.
                        (default: False)
```

-tags	Indica para printar/salvar os as tags das frases no arquivo txt. (default: False)
-resposta	Indica se deve buscar uma resposta. (default: False)
-verbose	Indica para printar/salvar todos os valores existentes. (default: False)
-quiet	Indica para não printar nada enquanto faz a execução. (default: False)
-original	Define que os resultados são referentes a frase passada por parâmetro. (default: False)
-final	Define que os resultados são referentes a frase final, depois de passar pelo modulo CBC e ser atualizada. (default: False)
--sem-resposta	Não vai mostrar a resposta mesmo se o parâmetro '-resposta' for passado. (default: False)
-somente-resposta	Indica que somente vai retornar a(s) resposta(s) corretas/finais. (default: False)