Pontifícia Universidade Católica
do Rio de Janeiro

Alysson Gomes de Sousa

An approach to answering natural language questions in Portuguese from ontologies and knowledge bases

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Profª Simone Diniz Junqueira Barbosa

Rio de Janeiro
December 2019

Pontifícia Universidade Católica do Rio de Janeiro

Alysson Gomes de Sousa

An approach to answering natural language questions in Portuguese from ontologies and knowledge bases

Dissertation presented to the Programa de Pós–graduação em Informática da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee.

Profª Simone Diniz Junqueira Barbosa
Advisor
Departamento de Informática – PUC-Rio

Prof. Hélio Côrtes Vieira Lopes
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Prof. Marco Antonio Casanova
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Prof. Antonio Luz Furtado
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Rio de Janeiro, December 13th, 2019

Alysson Gomes de Sousa

He graduated in Information Systems at the Federal University of Ceará (UFC). Interned at BTG Pactual Lab, developing Web front-end, and as database analyst at Data Science Lab (DasLab), both at PUC-Rio. Currently works as a Junior Researcher at the Pontifical Catholic University of Rio de Janeiro at DasLab Laboratory.

Bibliographic data

Gomes de Sousa, Alysson

An approach to answering natural language questions in Portuguese from ontologies and knowledge bases / Alysson Gomes de Sousa; advisor: Simone Diniz Junqueira Barbosa. – Rio de Janeiro: PUC-Rio, Departamento de Informática, 2019.

v., 100 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia

1. Ontologia – Teses. 2. Base de Conhecimento – Teses. 3. Web Semântica – Teses. 4. Processamento de Linguagem Natural – Teses. I. Diniz Junqueira Barbosa, Simone. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

This dissertation is firstly dedicated to my mother Raquel Moraes da Silva Sousa, to my father Almir Gomes de Sousa and my brother Arley Gomes de Sousa.

Acknowledgments

I thank God first for giving me the opportunity to study and work at the Pontifical Catholic University of Rio de Janeiro. I also thank my parents Almir Gomes de Sousa and Raquel Moraes da Silva Sousa, for all the teachings that shaped the person I am today and even my brother Arley Gomes de Sousa for their support.

I am very grateful to my advisor Simone Diniz Junqueira Barbosa, for all the knowledge she has given me and for all the opportunities entrusted to me, as these two years of masters would be much more difficult without her.

I thank Professor Hélio Côrtes Vieira Lopes for giving me the opportunity to work in the DasLab laboratory, where I can have experiences that have allowed me to mature as a researcher and professional.

I thank my friends Romulo, Micaele, André, and Dalai for receiving me in Rio de Janeiro and for supporting me throughout the master.

Finally, I thank my friends Raul, João Vitor, Luisa, Vinicius, Lauro, Sergio, Pedro, and Rodrigo for all their support during the master's degree. I thank all who directly or indirectly helped me on this path, may God bless you all.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq.

Abstract

Gomes de Sousa, Alysson; Diniz Junqueira Barbosa, Simone (Advisor). **An approach to answering natural language questions in Portuguese from ontologies and knowledge bases**. Rio de Janeiro, 2019. 100p. Dissertação de mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

In recent years we have seen the growth of the volume of unstructured data generated in the traditional Web. Therefore the Semantic Web was born as a paradigm that proposes to structure the content of the Web flexibly through domain ontologies and the RDF model, making computers capable of automatically processing this data, enabling the generation of more information and knowledge. However, to make this information accessible to users in other domains, there needs to be a more convenient way of looking at these knowledge bases. The Natural Language Processing (NLP) area has provided tools to allow natural (spoken or writing) is a convenient way to perform queries in knowledge bases. However, for the use of natural language to be useful, a method is required that converts a natural language question or request into a structured query. With this objective, the present work proposes an approach that converts a question/request in Portuguese into a structured query in the SPARQL language, through the use of dependency trees and structured ontologies in graphs, and that also enables the enrichment of question/request results by generating related questions.

Keywords

Ontology; Knowledge bases; Semantic Web; Natural Language Processing.

Resumo

Gomes de Sousa, Alysson; Diniz Junqueira Barbosa, Simone. **Uma abordagem para responder perguntas em linguagem natural na língua portuguesa a partir de ontologias e bases de conhecimento**. Rio de Janeiro, 2019. 100p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Nos últimos anos temos visto o crescimento do volume de dados não estruturados gerados na Web tradicional, e por isso a Web Semântica nasceu como um paradigma que se propõe a estruturar o conteúdo da Web de uma forma flexível, por meio de ontologias de domínio e o modelo RDF, tornando os computadores capazes de processar automaticamente esses dados e possibilitando a geração de mais informação e conhecimento. Mas para tornar estas informações acessíveis para usuários de outros domínios, é necessário que haja uma maneira mais conveniente de consultar estas bases de conhecimento. A área de Processamento de Linguagem Natural (PLN) forneceu ferramentas para permitir que a linguagem natural (falada ou escrita) seja um meio conveniente para realizar consultas em bases de conhecimento. Contudo, para que o uso da linguagem natural seja realmente efetivo, é necessário um método que converta uma pergunta ou pedido em linguagem natural em uma consulta estruturada. Tendo em vista este objetivo, o presente trabalho propõe uma abordagem que converte uma pergunta/pedido em Português em uma consulta estruturada na linguagem SPARQL, por meio do uso de árvores de dependências e ontologias estruturada em grafos, e que também permite o enriquecimento dos resultados das perguntas/pedidos por meio da geração de perguntas relacionadas.

Palavras-chave

Ontologia; Base de Conhecimento; Web Semântica; Processamento de Linguagem Natural.

Table of contents

List of figures

List of tables

O temor do Senhor é o princípio da sabedoria,
e o conhecimento do Santo a prudência.

Provérbios 9:10, *Bíblia Sagrada.*

1
Introduction

In recent years we have seen a staggering growth in the amount of data produced by digital media, reaching the order of zettabytes (ZB), according to the International Data Corporation (IDC). According to them, we will reach about 175 ZB in 2025 (Reinsel et al., 2018). However, much of these data is generated without any structure to support their automatic processing, making it impossible to interpret them by computational means. As an example, we can cite the data generated on social networks, blogs, and news sites.

To circumvent this problem, Berners-Lee et al. (2001) proposed the Semantic Web. It consists of an extension of the traditional web aiming to make web content processable by machines, where real-world entities mentioned on the web are understandable to computational agents (algorithms, search engines) (Berners-Lee et al., 2001).

The Semantic Web paradigm is based primarily on two technologies: domain ontologies and the RDF model. *Domain ontologies* are documents that describe the structure and entity relationships of a given domain, allowing multiple contexts to formalized into a processable framework by machines (Gruber, 1993). The *Resource Description Framework (RDF)* is a structure-independent metadata model that allows describing data of the most varied nature, enabling its automated processing (Lassila et al., 1998). With these technologies, it is possible to structure web data into computer-readable formats.

In addition to these technologies, advances in Natural Language Processing (NLP) research have also been significant allies in the task of structuring unstructured textual content on the Web, as the NLP area also aims to enable computational agents to understand human language. Thus, the Semantic Web and the NLP area have joined forces to help machines understand the vast content available on the Web.

Through Semantic Web methods, we can generate vast amounts of semi-structured data from which we can extract a large amount of information. This information can be useful to users of various domains, since the data are extracted from various sources on various subjects. From this arises the need to make these data more accessible to users.

To achieve this goal, the use of natural language for building queries proves to be an appropriate means, especially for users who do not know about computing or how to program (Capindale and Crawford, 1990). Thus, a major task is to convert a query or request for information in natural language into a structured query which, when executed, generates the correct answer to the question/request. This task presents the following challenge: how to perform this conversion process? In other words, how to capture the user's intention or desire expressed in a natural language question/request and translate it into a computationally processable query?

To deal with this problem of interpretation, the use of ontologies can be beneficial, because they provide a formal description of the entities and relationships specific of the domain, that may not be explicit in the question/request in natural language. Thus, the main research question of this paper is: *How to use domain ontologies to answer questions or request information in natural language (specifically in Portuguese)?*

Several systems and methodologies have been proposed to solve this problem. Among them, we can cite the works of Thanawala et al. (2014) and Dubey et al. (2016), who developed approaches for this problem in English, using general domain ontologies and lexical resources. We also highlight the work of Rodrigues and Gomes (2015), who developed a system that answers questions in natural language in Portuguese using some NLP techniques. In addition to these, other works such as those of Hakimov et al. (2013); Yao and Van Durme (2014); Li and Xu (2016) have used more sophisticated PLN features, such as dependency trees, which increase the expressiveness of consultations and allow the extraction of more precise relationships, when compared to the use of independent keywords.

In this work, we have developed an approach to answer questions or requests for information in natural language in Portuguese from ontologies that describe the domain, sophisticated NLP techniques, and knowledge bases built from these ontologies.

Besides the approach that allows answering the questions, we also propose a method to enrich the answer to the initial question, favoring the discovery of new information through the automatic generation of related questions.

To assess the effectiveness of the work, it was evaluated through a benchmark, created by the author himself, and through empirical evaluations with users.

This document organized as follows: Chapter 2 provides the rationale for the main concepts of the work introduced in the previous paragraphs. Chapter 3 presents related work on the Semantic Web and Natural Language

Processing. Chapter 4 details the approaches proposed in this paper, and Chapter 5 shows our evaluation methodology, followed by Chapter 6, which features the results of the evaluations. We conclude with Chapter 7, which presents our final remarks and future work.

2
Theoretical Foundation

In this chapter, we introduce the main concepts that underlie this work. In section 2.1, we cover the issues surrounding the Semantic Web; in section 2.2, we discuss natural language processing; Finally, in section 2.3, we discuss how the areas fit together.

2.1
Semantic Web

In recent years we have been following the staggering growth in the volume of digitally produced data. A survey by the International Data Corporation (IDC) estimated that in recent decades, we have produced data in the order of zettabytes (ZB), and it estimated that by 2025, we will have produced about 175 ZB (Reinsel et al., 2018).

Much of these data is used on personal computers, smartphones, different sensors, services, and other media that are connected to the Internet and stored in large data centers, so some of this data is accessible on the Web and used for a variety of purposes.

However, much of these data is generated to be processed by humans, that is, read and interpreted by people. This is evident by the lack of structure in the generated data. Taking content available on blogs, news sites, and social networks as an example, we see a large amount of textual data generated without structure or standardization. This makes automatic processing highly expensive or ineffective, because web-based search algorithms can only manipulate and reference the structure external to the searched content (for example, the tags that structure an HTML document), but do not understand the content itself.

To enable computational processing, structured data and rules are required to reason about this data. This has been done for years in the field of Artificial Intelligence, which has always studied ways to structure and represent knowledge. However, even when data are structured, systems often use a representation of a portion of the knowledge in a given domain without representing and sharing the same definitions with other systems.

Given these problems related to the lack of structure in data and rules and their low sharing between systems, in May 1994, at the first international conference on the World Wide Web (WWW), English physicist Sir Timothy John Berners-Lee addressed the need for semantics on the web (Berners-Lee, 1994). This need prompted him to publish an article on the subject seven years later, entitled "The Semantic Web: A New Web Content Format That Has Meaning for Computers Will Start a Revolution of New Opportunities" (Berners-Lee et al., 2001).

The Semantic Web is a traditional Web extension that aims to make Web content processable by machines so that the meanings of real-world entities expressed on the Web can be understood by computational agents (data mining algorithms, search mechanisms, etc.). This allows these agents to offer more sophisticated functionality, while also taking advantage of the diversity of information available on the large computer network. A peculiar aspect of the Semantic Web is its universality, as both data and structure must be universal within the context of the Web, sharing concept definitions. This brings one of the main challenges of this research area:

> "to provide a language that expresses both data and rules for reasoning about the data and that allows rules from any existing knowledge-representation system to be exported onto the Web" (Berners-Lee et al., 2001).

Meeting this challenge requires technologies that are flexible enough to support equally flexible representations. Two technologies have been widely adopted in this context: the eXtensible Markup Language (XML) and the Resource Description Framework (RDF). These technologies have been used both to describe knowledge representation models (such as domain ontologies) and to structure the data themselves.

2.1.1 Domain Ontologies

Among the models of knowledge representation, domain ontologies have been popularized as a flexible model for organizing the information and rules needed to reason about data (Berners-Lee et al., 2001).

The term *ontology* was born to describe an area of research in philosophy that studies the nature of being, of existence, and its implications (Sowa, 1994). In the context of Computer Science, the term ontology is used to define an explicit specification of a conceptualization (Gruber, 1993). In other words, an ontology is a document that contains the formal description of knowledge of a particular domain. This description is important because real world elements are only known within a system if there is a description of these elements, *i.e.*, a formalization of their existence within the system, *e.g.*, through an ontology.

In an ontology, formally declared real-world objects form the *universe of discourse*, where these objects have relationships with each other, which reflect the domain vocabulary and thus the knowledge the system will have about that domain (Gruber, 1993). Similarly, an ontology can be understood as a conceptual schema of a database, where tables have attributes and relationships, just like the objects described in the ontology.

The main advantages of using ontologies are the ability to share knowledge and the guarantee of internal consistency (Gruber, 1993). Due to their flexible nature, ontologies allow the knowledge of a domain to be extended through references to other ontologies, expanding the system's universe of discourse. This is one of the features that make the use of ontologies attractive in the context of the Semantic Web, as one of the biggest challenges is developing a domain description that is reused on the Web (Berners-Lee et al., 2001). And, as an ontology is, by definition, a formal description, this formality allows us to evaluate what has been described through algorithms, testing for constraints that cannot be satisfied or circular relationships, for example, guaranteeing its internal consistency.

With the maturation of this kind of representation, specific languages emerged to describe ontologies, among which the best known is the *Web Ontology Language* (OWL). OWL is a logic-based language that allows us to describe formalized knowledge in ontology, as well as facilitate consistency check and knowledge inference. Documents represented in OWL can also be exported to other formats, such as XML (McGuinness et al., 2004).

In short, domain ontologies are an attractive tool for expressing knowledge of a particular domain, allowing knowledge to be consistently and reusably described.

2.1.2 Resource Description Framework

With ontologies, we have a tool to describe knowledge of a particular domain. In addition to this description, a means is also needed to structure the content itself. For this purpose, the Resource Description Framework (RDF) was devised. RDF is a model that provides the foundation for metadata processing, making it easy to automatically process web resources, making them understandable to machines (Lassila et al., 1998).

The RDF model is fundamentally composed of triples in the form of subject, predicate, and object. The triples can be understood as phrases that describe a particular object. For example, in the phrase "Alysson is the author of this work," the subject (*Alysson*) is associated with the object (*this work*) through a predicate (*is author of*). Thus, RDF becomes a natural way of describing features that do not have a well-defined structure, such as natural language texts.

On the Web, all resources are indexed by *Universal Resource Identifiers* (URIs). This benefits the RDF model because both the subject, predicate, and object of a triple can reference resources through your URIs. Taking the previous example, Figure 2.1 shows how it could be structured using URIs: the URI associated with the entity "Alysson" is in the subject, the URI that references the entity "work" is in the object, and the URI associated with the authorship of the object (from the Dublin Core[1] ontology) is in the predicate that links the subject to the object.

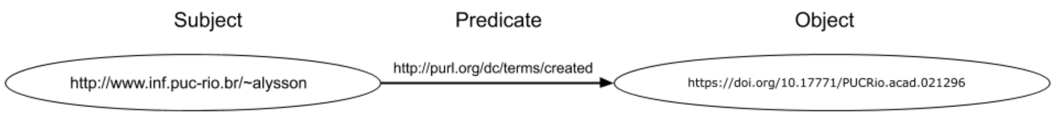

Figure 2.1: Triple RDF Example

Another essential feature of the RDF model is that it is graph-based (Lassila et al., 1998), as can be seen in Figure 2.1. This allows using a variety of graph theory algorithms and concepts: the resources (subjects or objects) identified by the URI represent nodes within the network, linked by several types of properties (predicates), which represent edges. Also, since RDF is a data model, it can be implemented in several languages. Currently, the most well-known implementations are in XML[2], JSON[3], Turtle[4], and N-Triples[5].

The RDF model also has a close relationship with domain ontologies, as it is the ontologies that define the vocabulary and macrostructure used in the triples. The Dublin Core ontology, used in our example, describes relationships between entities that represent scientific work, among other things. From this description, we are aware of what types of entities can be referenced in this ontology (people, institutions, books, articles) and how they can relate to one another (authorship, contribution, editorial).

[1] `http://www.dublincore.org/specifications/dublin-core/dcmi-terms/2012-06-14/`
[2] `https://www.w3.org/TR/rdf-syntax-grammar/`
[3] `https://www.w3.org/TR/rdf-json/`
[4] `https://www.w3.org/TR/turtle/`
[5] `https://www.w3.org/TR/n-triples/`

2.1.3 SPARQL

Given this new data structuring paradigm, an RDF structured data manipulation and query language, called SPARQL, was developed, which allows for the flexibility of the data model (Harris et al., 2013). As an example, consider the graph shown in Figure 2.2. Some nodes refer to instances, indicated by circles, such as *:alice*, *:bob*, and *:bach*, as well as other primitive data, such as texts and dates. Properties link all nodes, indicated by edges.

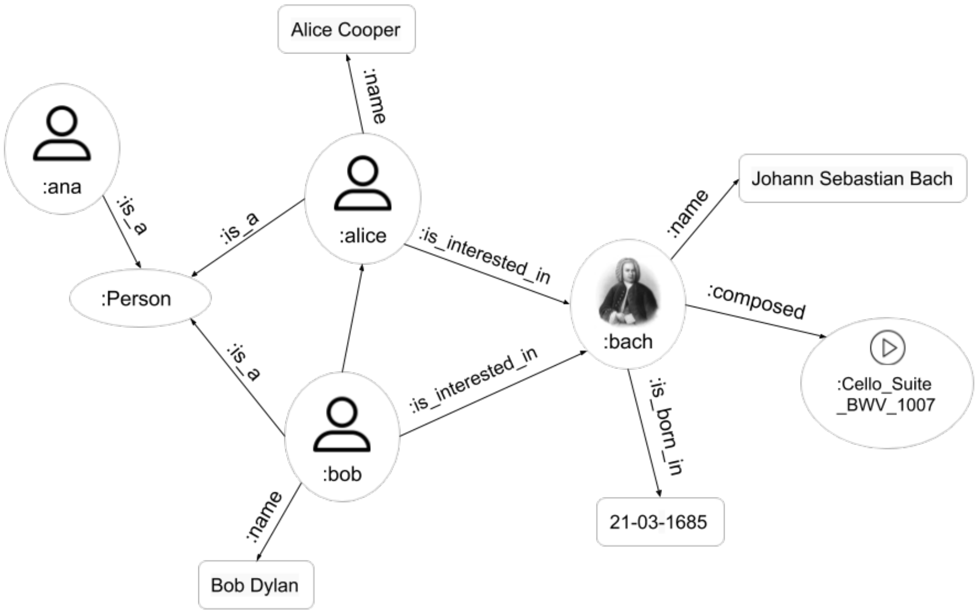

Figure 2.2: Exemplo de Grafo RDF

Code 2.1 is an example of a SPARQL query that lists the name of the instances that the people in the graph in Figure 2.2 are interested in (*is_interested_in* property). In this case, the result of the query will be: *Johann Sebastian Bach*.

```
SELECT ?name
WHERE {
    ?people :is_a :Person
    ?people :is_interested_in ?some
    ?some :name ?name.
}
```

Listing 2.1: SPARQL Query Example

Thus, the RDF model, supported by the SPARQL language, proves to be a very viable option for making web data understandable to machines, being primarily supported by domain ontologies.

2.2 Natural Language Processing

As stated earlier, there is a vast amount of data on the Web, much of which is in an unstructured format, making it difficult to process. To address this problem, the Natural Language Processing (NLP) area has been a great ally.

NLP is an area of Artificial Intelligence that seeks to extract a complete representation of meaning from free text (Kao and Poteet, 2007). To achieve this goal, NLP makes use of linguistic concepts, such as morphosyntactic classifications and grammatical structure, the lexicon of words and their meanings, their properties, grammatical rules, synonyms, and abbreviations, sometimes implemented in domain ontologies (Kao and Poteet, 2007). Some NLP tasks have become highly relevant to other areas of knowledge, including the Semantic Web. Among these tasks, we can highlight Named Entity Recognition (NER) and Relationship Extraction (ER). NER seeks to detect mentions of certain classes of objects in a text, such as people, organizations, or locations, and ER seeks to determine what type of relationship unites these entities within the text.

NLP has matured a lot in recent decades, evolving in terms of research and methods. Currently, the statistical approach has grown and consolidated, proving to be the best way to deal with the difficulties in processing natural language (Manning et al., 1999). As proof of this, the neural networks, which are statistical methods, have obtained meaningful results, mainly in NER and ER.

However, there are still many challenges in NLP that strongly influence the knowledge structuring process on the Web, the main one being ambiguity. Ambiguity occurs when a term or term set has more than one possible interpretation, significantly affecting the outcome of NLP algorithms (Manning et al., 1999). In the context of the Semantic Web, ambiguity can also result in processing errors. For example, some terms in a set of sentences may match more than one object in an ontology. In this case, it is necessary to define which object to use for each term. Other challenges that involve NLP, such as the need to determine the structural relationship of texts, even before determining the relationship between entities, are also highly relevant to other areas.

In NLP, dependency analysis is an essential task, as it seeks to capture the syntactic structure of the text. This structure is described in terms of sentence words and a set of directed binary grammatical relationships between words (Jurafsky and Martin, 2014). Figure 2.3 illustrates an example of a dependency structure (in Portuguese) based on the previous definition.

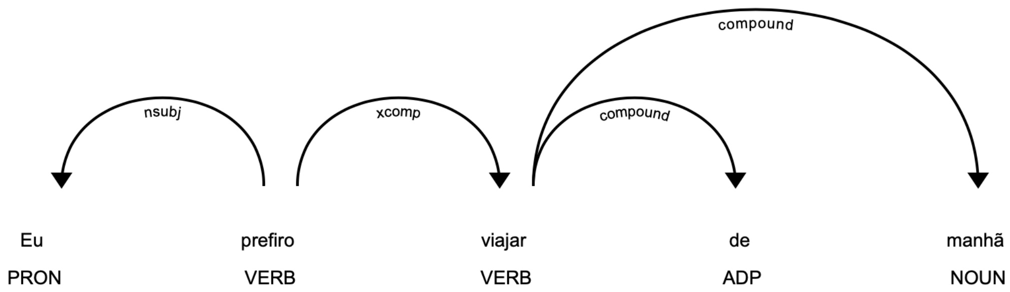

Figure 2.3: Dependency Tree Example

As we can see in the figure, the relationships between words are directed, starting from a term called *head* to another term, called *dependent*. Relationships have specific labels that indicate the type of relationship that exists between terms. This leads to the structure also called a *typed dependency structure* (Jurafsky and Martin, 2014).

The idea of dependency relations comes from the very notion of the *grammatical relationship* of traditional linguistics. However, linguists have developed several taxonomies for these types of relationships, with considerable variation between them. Despite the variations, an effort was made to develop a computationally useful standard, from which the *Universal Dependencies* project was born, which provides a set of linguistically-based standard relationships applicable to multiple languages (Nivre et al., 2016). These standard relationships used in the example in Figure 2.3, and Table 2.1 show a sample of relationship types.

Table 2.1: Sample of relationship types extracted from De Marneffe and Manning (2008)

Type relation	Description
root	root
det	determiner
nsubj	nominal subject
mod	modifier
nmod	nominal modifier
case	prepositions, postpositions and other case markers
obj	indirect or direct object
nummod	numeric modifier
conj	conjunct

One of the main advantages of using this dependency framework is that the relationships extracted in the analysis provide an approximation to semantic relationships. This approach is especially useful for applications within the context of the Semantic Web, as these relationships may indicate properties that syntactically linked to particular objects whose semantic relationship mapped in an ontology or any other representation of knowledge.

In addition, dependency analysis allows you to list terms that are normally processed independently, a common approach in keyword-based search services. Therefore, considering possible relationships between keywords can increase query expressiveness and yield more accurate results.

In short, these dependency structures are useful for extracting structured semantic relations from unstructured texts.

2.3 Answering Questions from Knowledge Bases

As we have seen, domain and RDF ontologies help us structure the knowledge contained on the Web, and the NLP area provides us with resources to make the computer understand what humans write. These two areas converge in support of various tasks. In this work, we focus on the task of answering questions asked in natural language.

The justification for this convergence lies in the availability of information. The Semantic Web seeks to structure unstructured Web data so that more abundant information can obtain that can be useful to practitioners in various fields of knowledge (medicine, biology, chemistry). However, obtaining this information requires knowledge of query and data processing techniques that are not common to these areas. Using natural language as a query method is best suited to users without computer skills (Capindale and Crawford, 1990; Kaufmann and Bernstein, 2007), so using NLP helps us solve the problem.

The integration between the areas occurs through the conversion of questions or requests for information into SPARQL queries, and it is in that conversion process that we face the main challenge of this task. What makes the conversion process challenging is the difficulty in capturing the intent behind the question or request, as there are several ways to ask the same question or make the same request (Höffner et al., 2017).

Another challenging aspect of this task, which was also mentioned earlier, is ambiguity. Ambiguity can manifest itself in a variety of ways, either syntactically or semantically, which strongly impacts the conversion of a question or request to a SPARQL query and may result in wrong answers (Höffner et al., 2017). Also, the very complexity of questions makes the process even more difficult, as complex questions may require filters, groupings, and even nested queries that are difficult to infer. Another challenge is the diversity of languages, which often differ significantly in syntax and semantics (Höffner et al., 2017).

Therefore, even joining the strengths of the Semantic Web and NLP, the problem is still challenging in many ways, which justifies this research effort.

3
Related Works

As we saw in the previous chapter, the problem we are addressing is complex and multifaceted. This motivated the development of many research works that have tried to solve the problem. In this chapter, we will list some of the relevant works, highlighting the relationships with our work.

3.1
Answering questions in natural language

There is a wide range of approaches for dealing with this problem. Many of these approaches use standard features such as NLP techniques (POS-Tag, Tokenization, Stemming, Lemmatization) and lexical features that relate words to their synonyms. In many cases, they use the same reference ontology, the DBpedia's ontology[1].

Among the works that fall into this context, we highlight those that, in addition to using DBpedia's ontology, also used the same lexical resource, called WordNet[2] (Thanawala et al., 2014; Dubey et al., 2016). These works differ in the method of query construction, as the work of Dubey *et al.*, which normalizes the question into a preconceived structure and generates the query from *templates*, while the work of Thanawala *et al.* is based only on the correspondence between the terms of the questions and the ontology. These works relate to ours in that they propose a generic approach and maximize the ontology's ability to detect concepts using lexical resources. This strategy will be considered in this work.

In addition to lexical resources, many works invest in some alternative representation for the data used, and a widely used representation is the vector representation. Vector representation consists of associating a term or term set with a set of numeric values (vector). The works of Berant and Liang (2014); Hartawan et al. (2015); Cortes et al. (2018) use vector representations to characterize data, but the way in which the representation is used differs significantly between works. In Berant and Liang's work, the vector representation is used to create a canonical form for the questions, allowing

[1]`https://wiki.dbpedia.org/services-resources/ontology`
[2]`https://wordnet.princeton.edu`

them to answer other questions if there is variation about the canonical form. In the works of Hartawan *et al.* and Cortes *et al.*, the difference is the measure of similarity used to find the answer: Hartawan *et al.* use measures DF-IDF (Document Frequency and Inverse Document Frequency) and TF-IDF (Term Frequency and Inverse Document Frequency), which determine the importance of the document and the importance of a term within a database, respectively; Cortes *et al.* also use TF-IDF, but together with an additional vector representation model that captures semantic data information, *Word Embeddings*. These works relate to this work because they propose a generic model to answer questions in natural language. We highlight the work of Cortes *et al.*, who proposed a model to answer questions in Portuguese, which our work also proposes to do.

Other works have proposed more specific approaches applied to specific contexts. This is the case of the AQUEOS system, developed by Toti (2014), which also uses NLP resources, but whose focus is on sentences with a single main verb. To generate the SPARQL query, the system indexes the concepts described in the ontology that are associated with the biomedicine domain. The system then looks for some mention of these concepts in the question and determines the type of answer that should be returned (a list, a numeric or alphanumeric value, or a reference to an individual). This work is related to ours because it also adopts a method that organizes ontology to assist in constructing the query.

Finally, Rodrigues and Gomes (2015) created a system that answers questions in Portuguese from the data of the CHAVE[3] corpus, where the authors focus on the identification of named entities by reducing terms to their lemmas, and consequently, the construction of search triples is defined by entities that may be in the subject or object of the search triples. This work is related to ours because he also set out to answer questions in Portuguese.

Although the last three works mentioned above are not intended to be generic, they came from a specific context, just like ours.

3.2 Answering questions using ontologies and dependency trees

Even though there is a wide diversity of approaches, in this section we focus on those based on dependency structures.

Some of the methods are based mainly on the sentence dependency tree, among which we can cite those by Hakimov et al. (2013); Yao and Van Durme

[3] `https://www.linguateca.pt/CETENFolha/`

(2014); Li and Xu (2016); Baudiš and Šedivỳ (2015).

Their works are guided mainly by dependency tree structure because, as already said in the previous chapter, the relationships of these trees indicate semantic relationships. Therefore, these relationships guide the methods that result in triple fetching in a SPARQL query. Differences in methods are about ontologies: Hakimov *et al.* and Li and Xu used DBpedia, but Yao, Van Durme, Baudiš, and Šedivỳ used FreeBase. These works are strongly related to ours because our method is also guided by the structure of the dependency tree, but not only by it.

Other works integrate dependency trees into other methods and resources. This is the case of the works by Yang et al. (2015); Paredes-Valverde et al. (2015). The first uses vector representations to capture lexical and semantic characteristics, in addition to the semantic relations captured in the dependency trees — these vectors used as canonical forms of properties that relate one or more mentioned concepts. The second proposes a system called ONLI, which uses trees together with an ontology-based question model and a question classification scheme proposed by the authors themselves.

Lopez et al. (2016) proposed a generic method that uses named entity detection to identify concepts and dependency trees to relate them. However, it integrates other dependency structures to build the query that will generate the response. They developed a method for querying information in an educational domain ontology, which is mostly based on a dependency tree, which translates the questions in Portuguese into SPARQL queries. The method does not use lexical resources to expand term detection capabilities and is limited to factual and definition issues.

These works have a strong relationship with ours because we also integrate the detection of semantic relationships through dependency trees with the indexing of the reference ontology.

3.3 Disambiguation of concepts

As stated in the previous chapter, ambiguity is a recurring problem, and in systems that answer questions in natural language, this problem significantly impacts the quality of the answers generated. Therefore, many systems choose to assign the user the task of determining the correct interpretation in situations where there is ambiguity.

Melo et al. (2016) developed a cooperative dialog manager that answers user questions by structuring questions with a discourse representation framework, identifying concepts through word similarity to DBpedia's ontology concept labels, expanded with lexical features such as WordNet. We highlight its disambiguation method because it is user-oriented: as the concepts coincide with more than one question term, the user is notified, and the system asks them to choose one of the possible interpretations.

Damljanovic et al. (2012) and Kaufmann et al. (2006) developed tools called FREyA and Querix, respectively, which rely on entity detection via character similarity and dependency trees to relate them. However, the disambiguation is up to the user, as the systems generate a dialog box for the user to tell the interpretation for ambiguous terms, either concepts or attributes.

These works are strongly related to ours because we will also use user feedback to determine the interpretation of ambiguous terms.

4
Methodology

In this chapter, we present the details of our proposal, which mainly consists of creating a method that can answer natural language questions in Portuguese from a domain ontology and a knowledge base modeled from the ontology.

4.1
Reference Ontology

As we saw in Chapter 3, to develop an approach that answers questions from a knowledge base, we can use an ontology that models it. In our case, we use an ontology that describes the domain of oil and gas production.

This domain was chosen as use case because of a project we are working on, whose goal is to extract knowledge from textual reports. But all evaluations performed in our approach were done in the cinematic domain using an ontology of films.

To describe this domain, we built an ontology-based on ISO 14224 (14224:2016, 2016), which describes all equipment and maintenance and failure events involved in oil and gas production, so all terms used in this chapter come from that ISO norm. Figure 4.1 shows a hierarchical view of the main classes of ontology.

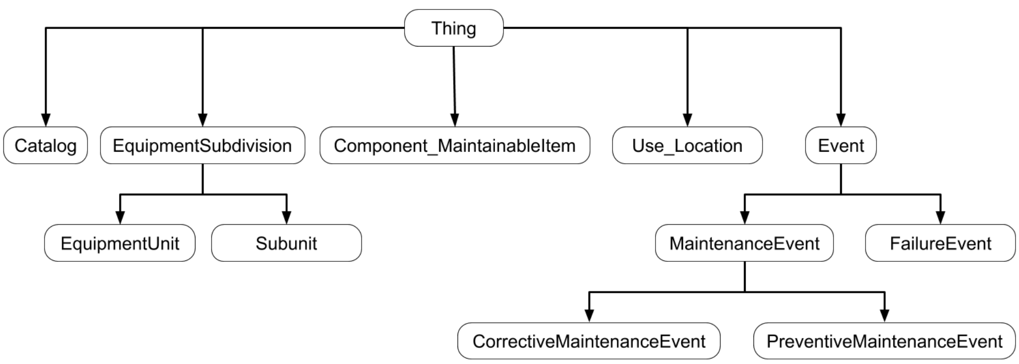

Figure 4.1: Principal classes of ontology

As shown at the first level, five classes constitute the main hierarchies of ontology. The *Event* class is responsible for representing the major types of events described in ISO 14224: failure (*FailureEvent*) and maintenance (*MaintenanceEvent*) events. In maintenance, there are also two essential classes, which represent the hierarchy of maintenance events: Corrective Maintenance (*CorrectiveMaintenanceEvent*) and Preventive Maintenance (*PreventiveMaintenanceEvent*).

The *Component_MaintainableItem* and *EquipmentSubdivision* classes represent equipment hierarchies and compositions, and just below the *EquipmentSubdivision* class are two other classes: *EquipmentUnit* and *Subunit*. These classes reflect the composition of the equipment since, according to the ISO standard, the equipment units are composed of subunits, which in turn are composed of maintenance items or components.

In addition to the events and equipment composition, there is the *Use_Location* class, which represents information about the context in which the equipment is being used, such as industry type, business category, plant, or platform where the equipment is installed.

Finally, we include a catalog (*Catalog*) that contains a hierarchy of classes that represent important additional event and equipment information, such as equipment states, fault detection methods, maintenance activities. In Figure 4.2, we have a more detailed view of the ontology, where dashed lines show the main properties that relate the classes.

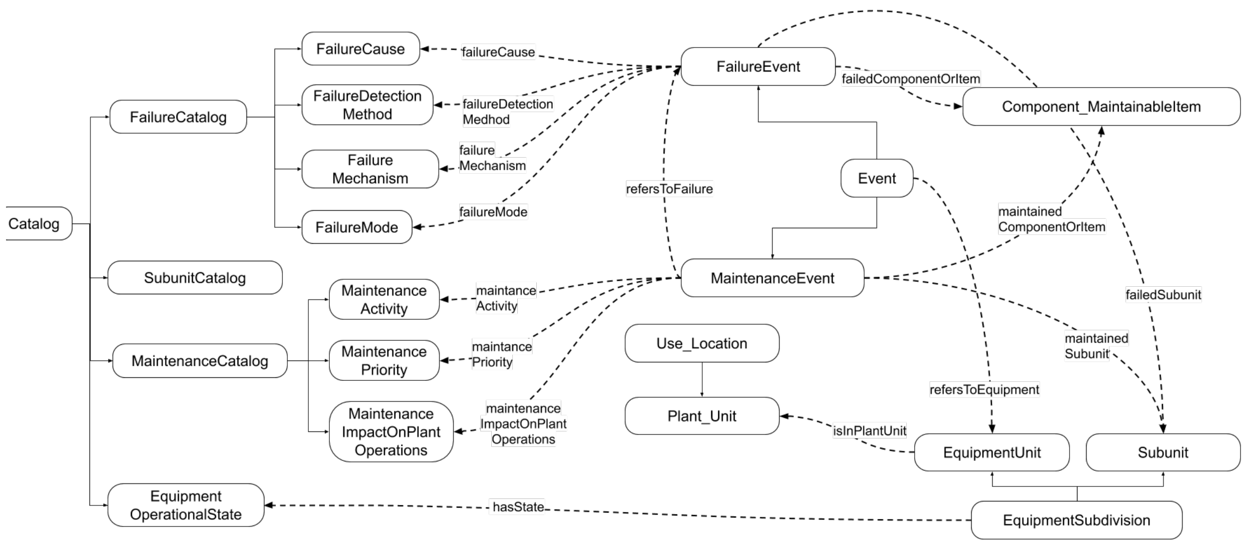

Figure 4.2: Relationships between the main classes of ontology

This ontology has many other classes and properties that describe each of the classes, but it is not relevant to describe them here. The important thing to note is that ontology offers a reasonable set of classes and properties that will support the development of the approach.

4.2 Interpretation mechanism

As we mentioned earlier, the goal of this project is to create a mechanism that is capable of answering natural language questions in Portuguese based on an ontology and knowledge base. In other words, this mechanism must be able to capture the intention or desire expressed in a user question or request and convert it into a SPARQL query.

To perform this process of interpretation, some steps also need to be performed: the detection of entities (section 4.2.1), the disambiguation of these entities (section 4.2.2), the ontology indexing process (section 4.2.3), and the extraction of relationships between entities (section 4.2.4). In the next sections, we will describe the implementation details of each of these steps.

4.2.1 Entity Detection

The first step in the interpretation process is entity detection. This detection step consists of identifying the classes, properties, and individuals expressed in the reference ontology and knowledge base mentioned in the question or request for information.

To accomplish this task, we first assume that all classes, properties, and individuals are annotated with the *label* property, defined in the standard RDF[1] vocabulary, preferably set to Portuguese (consisting of adding the suffix *@pt-br* after the *label* content). In addition, it was necessary to deal with variations in terms (due to verbal inflections, gender changes, and word grade). For this, we extract the radicals of the words; thus, the detection takes place by extracting the radicals of the terms of the question, and comparing the *n-grams* of these radicals with the radicals of the terms of the ontology *labels*.

For example, in the sentence "*Quais centrífugas estão inoperantes?*" we detect the terms *centrífugas* and *inoperantes* as entities present in the ontology vocabulary, since the *centrífug* and *inoper* radicals coincide with the same radicals present in the *label* of the class representing a centrifuge and on the *label* of the individual representing the former state in the ontology, respectively.

We adopted an organization strategy similar to the Paredes-Valverde et al. (2015) question model, where their types separate the detected entities. Thus, classes, properties, and individuals are in different groups but associated with the terms of the question or request. We are also using a classification of questions similar to Paredes-Valverde *et al.*'s, in order to capture the intent

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

of the user's question or request. In other words, the rating serves to evaluate whether the user wants an actual value, a set of values, dates, names, or any other type of resource.

To improve entity detection capabilities, we use a lexical feature called Onto.PT, created by Gonçalo Oliveira and Gomes (2014). This feature consists of a synonym ontology similar to WordNet but developed for Portuguese. This feature adopts the concept of synsets, which are sets of *synsets*. With this feature, we will evaluate whether the synonyms of the terms of the question or request are contained in the ontology vocabulary if the terms themselves are not present.

4.2.2 Disambiguation

Depending on the reference ontology, it is common to use the same terms to designate different entities. In our context, this is no different, so a disambiguation step is required.

The disambiguation step checks which terms or sets of terms match more than one entity. This includes the classes or properties of the ontology itself and individuals expressed in the ontology or knowledge base. After identifying these ambiguous entities, we pass these terms to the user with their respective interpretation options, so that they determine which option is correct.

To exemplify, still in the same sentence "*Quais centrífugas estão inoperantes?*", the term *centrífugas* is associated with more than one type of centrifuge (centrifuge as a subunit of equipment or as a maintenance item in a subunit), so the user must tell the type of centrifuge to consider.

4.2.3 Ontology Indexing

In addition to the previous steps, we have included a step that is performed offline, which is ontology indexing. This step arose from the need to manipulate the ontology more conveniently. In Figure 4.1, we see that the ontology consists of a tree that describes the hierarchical relationship of classes, and properties, while not explicit in the hierarchy, relate classes that are at the same or different levels.

So the idea of indexing an ontology is to create a global graph that unites the hierarchy, individuals, and relationships expressed in properties. Figure 4.3 shows our indexed reference ontology in the form of a graph, where black edges indicate hierarchy edges and blue edges indicate properties.

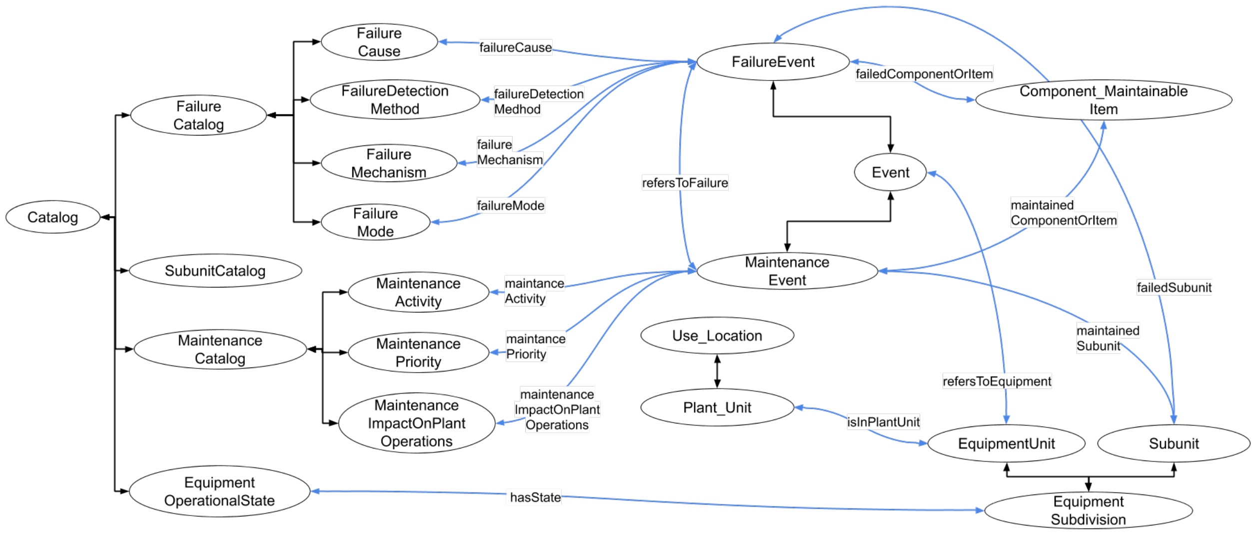

Figure 4.3: Indexed Ontology

To improve the search for paths within the graph, we create new properties that create the reverse direction of properties that do not have their corresponding inverse property (for example, hasActor and isActorIn, indicated by OWL's InverseOf[2] property), making all edges bidirectional.

The advantages that motivate this form of indexing are mainly two: (i) the ability to navigate more than one ontology as long as they connected in some way, which is extremely useful in general-purpose ontologies and usually reuses the structure of other ontologies; and (ii) the search for semantic relations, which consists in finding a path that connects two classes or finding a path that goes through a certain edge, and which is essential when there is a mention of some property that connects two classes.

Despite those advantages, a considerable disadvantage is the performance, as some tasks on large graphs require a lot of processing time. Particularly in our problem, this disadvantage has a minor impact, as the main activity is search, and the graph generated by the indexing process is sparse.

Currently, in our graph, we have 1758 nodes and 3658 edges. Considering a search algorithm like Dijkstra, whose worst-case complexity is $O(|E| + |V| \log |V|)$, as our graph is sparse, the performance is acceptable.

Therefore, this way of the structuring the ontology has proved to be very useful, not only in practical terms, such as support for the search for relationships, but also in theoretical terms, since the concepts of graph theory (such as shortest path and neighborhood) could also be used in this approach.

[2]https://www.w3.org/TR/owl-ref/inverseOf-def

4.2.4 Extraction of semantic relations

Finally, having all entities identified and the ontology adequately indexed, the final step is to extract the semantic relationships between the entities that were detected. The question guides this step or requests the dependency tree received as input. To describe the process, we will take the following sentence as an example: "Qual o sistema de compensação do guindaste que falhou mês passado?". Figure 4.4 shows the dependency tree corresponding to this question.

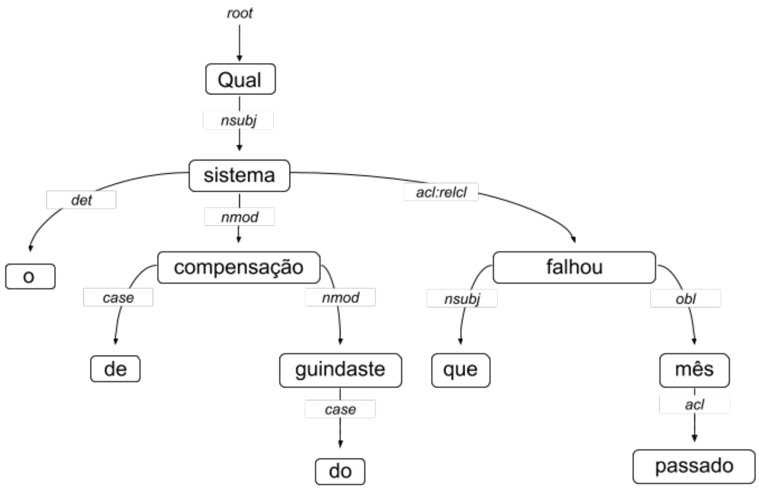

Figure 4.4: Sentence Dependency Tree

In our example, the terms *guindaste* and *falhou* coincide with reference ontology classes, and the term compound *sistema de compensação* corresponds to a property whose domain is the *guindaste* class. In addition to these elements, we must consider the terms *qual*, *mês*, and *passado*, as they expose essential details for the question interpretation process: the first informs us that the user wants a value or set of values, and the second and third describe characteristics — great storms for the user.

With this information, to simplify the processing and structure of the dependency tree, we perform a concatenation of the compound terms. Thus, terms such as *sistema de compensação* are transformed into *sistema_de_compensação*, and the tree is restructured, as shown in Figure 4.5.

From this new dependency tree, we extract the relationships. To accomplish this task, we walk the tree from its root, evaluating each of the nodes with their respective children and siblings. If we find a node that corresponds to a class, we propagate this information to the child and sibling nodes, so that the next evaluated nodes that match some class are related to the previous node.

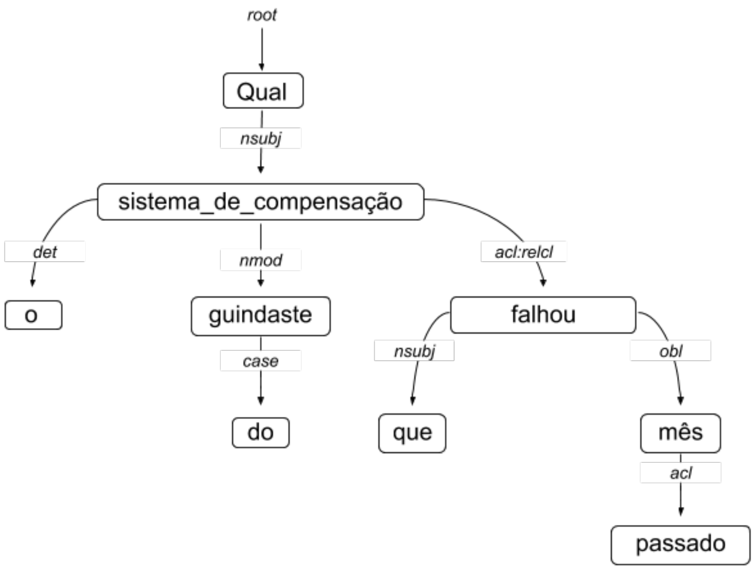

Figure 4.5: Restructured Dependency Tree

This relationship will be built from the path that joins the two nodes in the indexed ontology, resulting in query triplets, as illustrated in Figure 4.6.

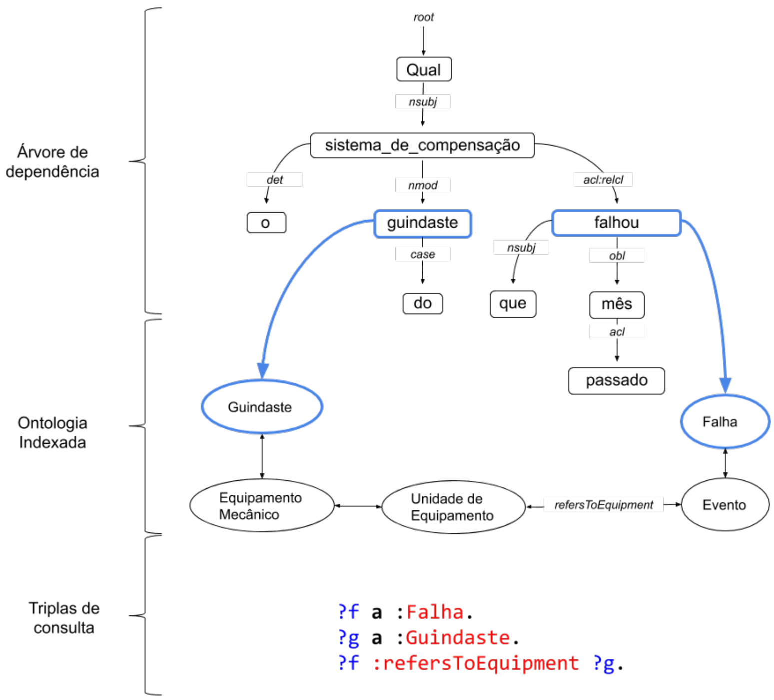

Figure 4.6: Relationship extraction from dependency tree

Another case that should be highlighted is the existence of references to properties of a type *object property*, which indicate relationships between certain classes. In this case, it is necessary to consider a path between the classes that necessarily passes through the edge indicated by the property, as shown in Figure 4.7. The figure illustrates the process of extracting sentence relations. "*Quais controles de backup são partes dos computadores da P-50?*".

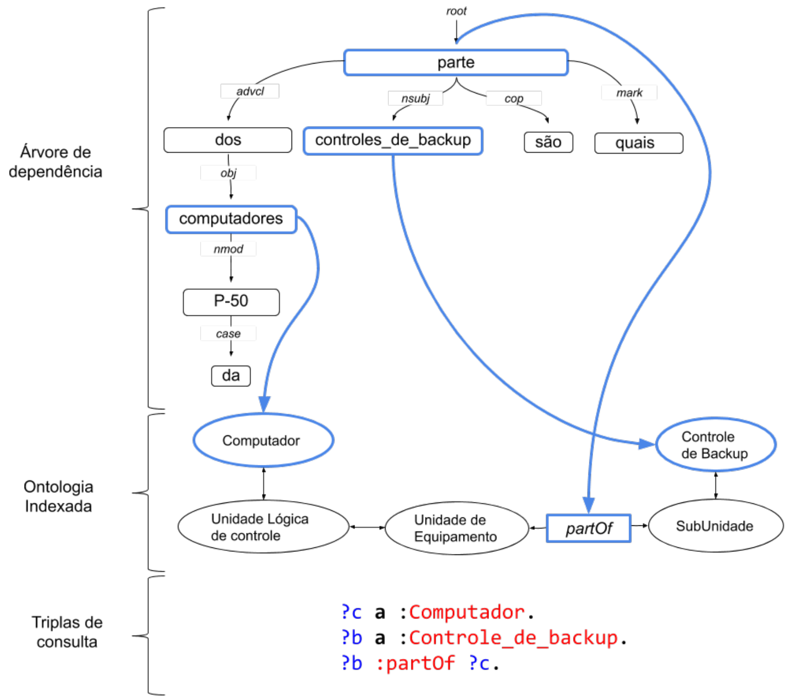

Figure 4.7: Extraction of relation to definition of a fixed edge

After extracting the relationships between classes, considering the existence of references to properties of the *object property* type, we also consider the existence of individuals expressed in the ontology itself. For these cases, we perform the same processing as classes, because individuals are directly bound to classes through the *type* property. The algorithms used for dependency tree navigation and conversion of the ontology graph path to query triplets can be found in Appendix A.

Finally, we evaluate the other terms that match the keywords defined in our method, where terms processed as parameters whose values are in the neighborhood of the term or its subtree. Tables 4.1 and 4.2 present the terms we consider as keywords and their meaning during the process of interpreting the question or request.

Thus, we conclude the relationship extraction process, considering everything mentioned in the ontology and what we consider relevant to create a query that corresponds to what the user expressed in their question or request.

Table 4.1: Keywords considered in our approach - Part 1

category	**keyword**	**meaning**
factual	qual/quais	select a value or a list of selected values.
	quantos	indicates the need for a count so that the result will be numeric.
	quando	indicates a value that matches a date or period of time.
	quem	indicates a value associated with a person or organization.
	onde	indicates a value associated with a place.
temporal	dia(s)	indicates the need for filtering that takes into account one or more days, and may be qualified by terms like *ontem*, *último* or *passado*.
	mês(es)	indicates the need for filtering that takes into account one or more months or a specific month, and may also qualify for terms like *último* or *passado*.
	ano(s)	indicates the need for filtering that takes into account one or more years, or a specific year, and may also qualify for terms like *último* or *passado*.
	bimestre(s)	indicates the need for filtering that takes into account a set of days corresponding to 2 months.
	trimestre(s)	indicates the need for filtering that takes into account a set of days corresponding to 3 months.
	semestre(s)	indicates the need for filtering that takes into account a set of days corresponding to 6 months.
	década(s)	indicates the need for filtering that takes into account a set of days of at least 10 years.
	data específica	indicates the need for filtering that takes into account a specific date, such as 04/05/2019.

Table 4.2: Keywords considered in our approach - Part 2

category	keyword	meaning
qualifiers	mais	Indicates the need for a descending sort of results.
	maior	
	menos	Indicates the need for an ascending sort of results.
	menor	
grouping	agrupado(a) por	Indicates the need for a grouping of the data by certain criteria.
	agrupada pela	
	agrupado pelo	

4.3 Related Question Generation Engine

In the search for information, it is natural that, initially, a user does not know how to formulate textually what they want to search, so they can start from an initial set of terms that will generate a result, and from this result, the user will refine the search to improve results, as pointed out by Marchionini (1997).

To optimize this process, many works, such as Sun et al. (2010); Setlur et al. (2016); Gao et al. (2015), have proposed models that provide information related to the initial results, in order to reduce the user's cognitive effort to formulate new terms or questions that refine or broaden the results of the search.

Therefore, in addition to the mechanism for interpreting natural language questions, our approach provides a way to enrich the first answer (gained by answering the first question) through relationships identified in the initial question and ranked by the strengths of those relationships.

4.3.1 Annotation Scheme

Like the interpretation engine, the related question generator also makes use of the reference ontology. The ontology is enriched with annotations that define relationships that the user finds interesting, given that certain entities were cited in the initial question.

To structure the annotation scheme, we created an ontology that defines the format of annotations. Annotations are first defined as an *Annotation*

Table 4.3: Annotation terms

term	**description**
hasRelationshipWithClass	indicates that a class has a relationship of interest with another class.
hasRelationshipWithNamedIndividual	indicates that an individual defined in the ontology has a relationship of interest with another individual.
hasRelationshipWithProperty	indicates that a property has a relationship of interest with another property.
isBaseCategoricalLevel	indicates a class hierarchy that can be used as related entities.
hasNotRelationshipWith	indicates that a particular entity does not have a relationship of interest with another entity.
relationshipStrength	indicates the strength of a relationship of interest.

Property defined by the OWL[3] vocabulary. This way, annotation ontology can be imported into reference ontology, and the user can define relationships of interest. Table 4.3 describes the terms defined in the annotation ontology. Because of the way annotation terms have defined, other ontologies may use them similarly to other annotation properties defined in the OWL vocabulary, except for the *relationshipStrength* annotation, which is defined as a *Datatype Property*, because it lists a list (as two related entities) with a numeric value.

In addition to these, we also created a property called *isBaseCategoricalLevel*, which identifies that a class hierarchy can be used as a set of relationships of interest; that is, for any class present in the hierarchy mentioned, its child classes could be used as related terms.

To exemplify the use of annotation, we will take an ontology that describes the film domain. In this ontology, we will have some elements, such as movies, TV series, awards. From these elements, a user can indicate which entities should be taken into account when others are mentioned.

Figure 4.8 shows the before-mentioned ontology with relationships of interest defined with their strength indicated in relationships, where green edges indicate relationships mapped with the *hasRelationshipWithClass*, *hasRe-*

[3]https://www.w3.org/TR/owl-ref/#Annotations

lationshipWithNamedIndividual, or *hasRelationshipWithProperty* annotation, whose value is set by the *relationshipStrength* property, already the edges red indicates invalid relationships that should not be considered, defined with the *hasNotRelationshipWith* property; finally, the classes with green outline represent the class hierarchies marked with the *isBaseCategoricalLevel* property.

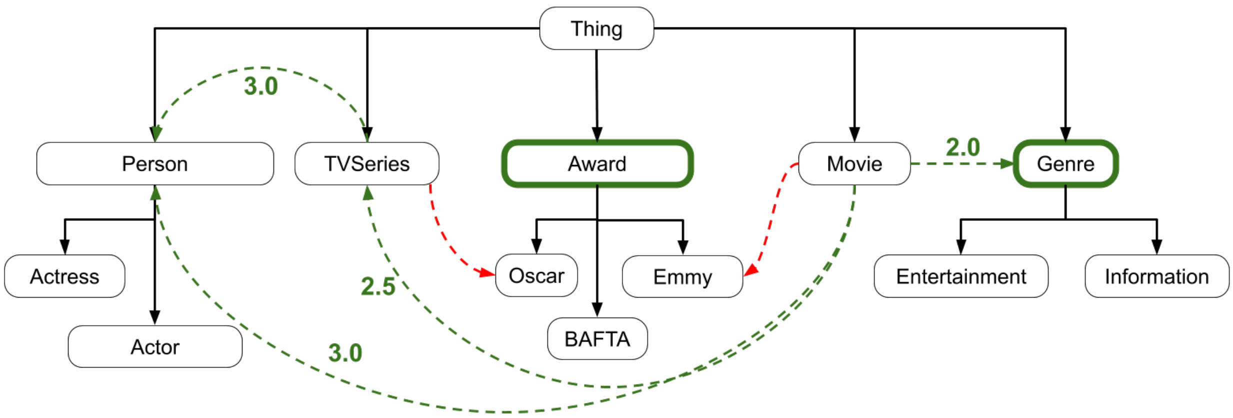

Figure 4.8: Annotated ontology

4.3.2 Question generation

Once we have the ontology appropriately annotated, our strategy for generating related questions will take notes to generate the questions.

The process consists of using the entities identified in the interpretation mechanism; from this point, we can rephrase the initial question replacing the entities mentioned by related entities (defined by the user at the time of annotation).

Figure 4.9 schematically shows a clipping of the previous ontology with their respective relationships of interest and an initial question with some identified ontology entities.

Given the initial question, the strategy is to generate valid combinations between relationships, that is, any combination that is not marked with the *hasNotRelationshipWith* annotation. So if we take the question from Figure 4.9, we can generate the following related questions:

- Quais atores mais receberam oscar no último ano?
- Quais atrizes mais receberam oscar no último ano?
- Quais filmes mais receberam BAFTA no último ano?
- Quais Gêneros mais receberam oscar no último ano?

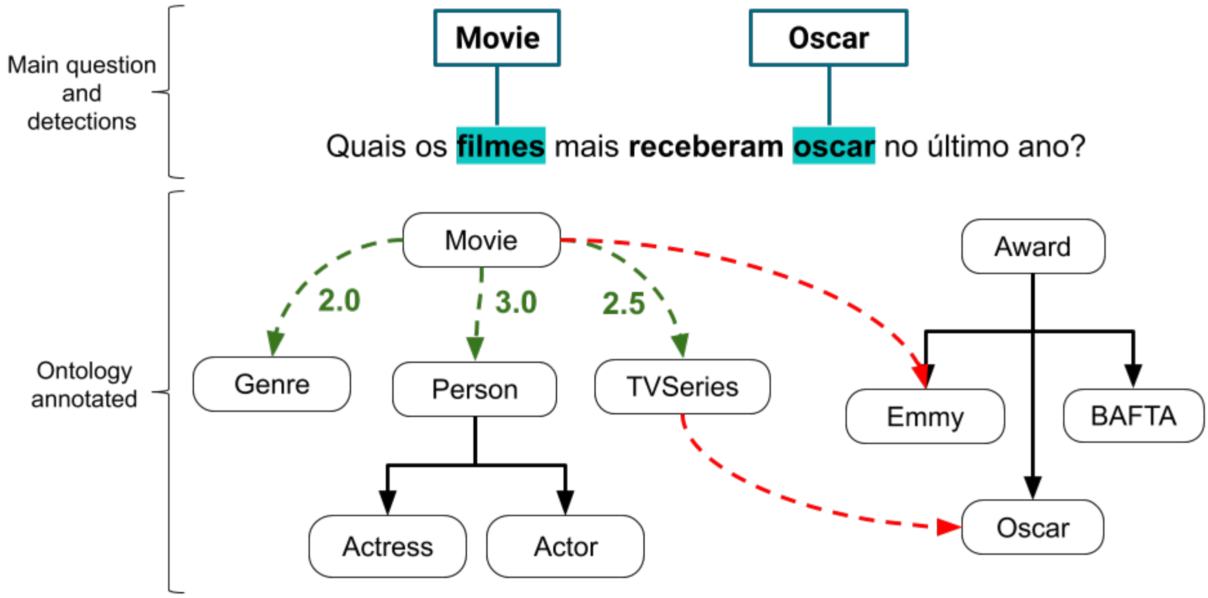

Figure 4.9: Sample Question and Its Relationships of Interest

– Quais Gêneros mais receberam BAFTA no último ano?

As can see from the examples above, combinations generate variations of questions that merge one or more entities, allowing the main question to expanded to the fullest.

However, because of the volume of questions generated, which is directly related to the number of entities cited in the initial question, we define an ordering criterion that combines the strength of the relationship (defined in the *relationshipStrength* property) with the number of different entities in each related question, defined by Equation 4-3.

$$C = \{r|\text{interest relationship generated by annotations}\} \tag{4-1}$$

$$w = \{\text{set of forces associated with relationships of interest}\} \tag{4-2}$$

$$s = \frac{\sum_{r \in C} \frac{1}{|C|} * w[r]}{|C|} \tag{4-3}$$

This criterion assumes that the higher the number of varied entities in the related questions, the farther it is from the initial question, so in Equation 4-3, the larger the size of C, the lower the strength of relationship s. of the question related to the initial question, even though the strength of each relationship, defined in w, is high. Thus, ranking related questions is initially made up of questions with few variations and high relationship strength to questions with many variations but with low relationship strength.

5
Evaluation Methodology

In this chapter, we describe how we evaluate each part of our work by considering objective aspects of the natural language question interpreter and subjective aspects of the related question generator.

5.1
Question Interpreter

Initially, we propose to evaluate our approach with the Question Answering over Linked Data (QALD[1]), held in 2018, developed based on BDpedia's data and ontology. However, the ontology has presented some problems that strongly impact our approach. First, approximately 40% of the properties of type *object property* do not have a *domain* or a *range* defined, and this information is particularly important because from them we define the edges of the ontology graph; In addition, about 15% of the *datatype properties* do not have a defined *domain*, making it challenging to identify class-specific properties.

We believe that these problems are related to the goal of the ontology, which seeks to describe the most varied domains generically, but this ends up making their use unfeasible by approaches that are guided by the structure of the ontology, as is the case of ours.

Because of these limitations, we decided to use an ontology that describes the film domain, developed at the Zurich University computer department, available at Github [2]. In addition to the ontology, they also developed a *plugin* for the Protégé tool, which supports the triple generation process from a relational database. Through this *plugin* we created a knowledge base from the IMDb database[3].

So, to evaluate our approach, we take the film domain ontology and knowledge base and build a question dataset based on the QALD competition question dataset, adapting the structure and questions to the IMDb context.

To build this mapping, we first take the main question types in the QALD dataset, such as questions that have the terms *what*, *who*, *when*,

[1]Training and Model Testing dataset is available on GitHub (`https://github.com/ag-sc/QALD/tree/master/9/data`)

[2]`https://github.com/ontop/ontop/wiki/Example_MovieOntology`

[3]`https://www.imdb.com/interfaces/`

where, then take the main operations applied to queries, such as count, sorting, grouping, and temporal filtering. Finally, we associate the classes and individuals mentioned in the QALD questions with the classes and individuals in the IMDb context.

Therefore, both datasets can be compared in terms of a variety of questions and queries structures. This mapping can be found in Appendix C.

Our dataset[4] is composed of 150 questions in Portuguese and English, with the SPARQL query of each question (based on the ontology we use) which we want the answer. The Listing 5.1 illustrates the structure of the dataset.

```
{
    "id":"1",
    "question_pt":"Quais os títulos dos 5 filmes de maior duração?",
    "question_en":"What are the titles of the 5 longest films?",
    "query":"
    PREFIX mo:<http://www.movieontology.org/2009/10/01/movieontology.owl#>
    SELECT ?title
    WHERE{
        ?m a mo:Movie .
        ?m mo:runtime ?runtime .
        ?m mo:title ?title .
    }
    ORDER BY DESC(?runtime)
    LIMIT 5"
}
```

Listing 5.1: Structure of dataset

In addition, we propose a baseline model that will be directly comparable to our approach. To measure the quality of the method, we will use *Precision*, *Recall*, and the F-score as metrics, defined as follows:

$$Precision(Q) = \frac{\#\ of\ correct\ answers\ of\ method\ for\ Q}{\#\ total\ method\ responses\ for\ Q} \tag{5-1}$$

$$Recall(Q) = \frac{\#\ of\ correct\ answers\ of\ method\ for\ Q}{\#\ total\ correct\ answers\ for\ Q} \tag{5-2}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5-3}$$

[4] Available at `https://github.com/alyssongomes/dataset-questions-imdb`

5.2 Related Questions Generator

To enrich initial user information, our approach is based on a semi-automatic strategy that relies on the human effort to define relationships of interest that will be automatically combined and returned to the user as questions.

Therefore, we take the IMDb ontology and define a set of relationships of interest and the strengths of each relationship, defined arbitrarily by our research group. But, we emphasize these relationships and their strengths can be defined manually or automatically in other cases. These definitions can be found in Table 5.1.

From that, we will evaluate the effectiveness of the related questions generator, in order to verify the quality of the questions generated from the combined relationships and the ordering criterion.

Table 5.1: Interest Relationships

Source	Target	Strength
Actress	Movie	3
Actor	Movie	3
TVSeries	Movie	3
Movie	TVSeries	3
Actor	Actress	3
isAwardedWith	nominatedFor	3
nominatedFor	isAwardedWith	3
Movie	Actress	2
TVSeries	Actress	2
Actress	Actor	2
Movie	Actor	2
TVSeries	Actor	2
Genre	Movie	2
Movie	Genre	2
Actress	Genre	1
Genre	Actress	1
Actress	TVSeries	1
Actor	TVSeries	1

The evaluation will be performed from the perspective of users who have some familiarity with search engines. For this, the user will evaluate the questions generated from the following set of predetermined sentences:

- (P1) Quais as séries de TV mais bem avaliadas no IMDb em 2018?
- (P2) Quais os 5 filmes que tiveram as maiores receitas bruta?

– (P3) Quais os 5 filmes de maiores durações?

These questions start the generation of new related questions, which can be found in Appendix B.

To evaluate these questions, the user will inform how much they consider that the generated question is related to the initial question, within a 7-point scale (1-Not related to 7-Strongly related).

To evaluate the order in which questions are reported, participants receive an online form with 2 question sets, P and A, where P contains the related question groups P1, P2, and P3 in the order proposed by our approach and A contains the same groups of question listed in random order (which convenience we will call A1, A2, and A3 respectively). To reduce the learning effect, we formed two user groups, each one receiving the question groups in a different order, as shown in Table 5.2. At the end of each question set, the participant evaluates the set of related questions as a whole and the order in which the related questions were listed, and chooses their preferred group.

Table 5.2: Participant Groups

Group	Question Order
G1	P1, P2, P3, A1, A2, A3
G2	A1, A2, A3, P1, P2, P3

Figure 5.1 shows a cutout of the digital form sent to participants.

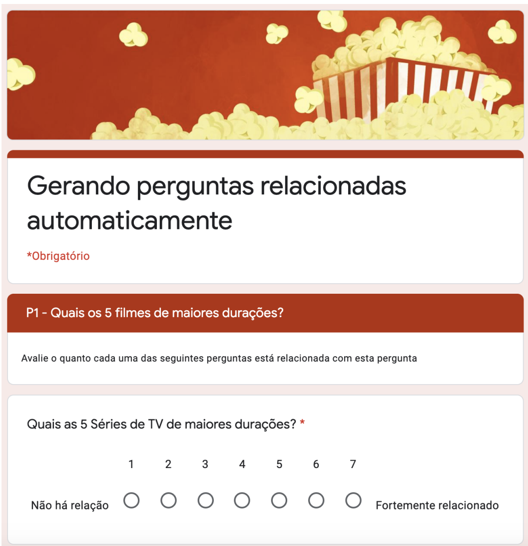

Figure 5.1: Digital Form

In the next chapter we will present and discuss the results with the execution of the experiments.

6
Results

In this chapter, we present and discuss the data and results obtained from the evaluations of the question interpretation engine and related question generator.

6.1
Dataset and Knowledge base

As stated in the previous chapter, we built a dataset from the ontology proposed by the Zurich University research group, and from the data available in the IMDb database. To guide the construction process, we used the QALD 9 question dataset. Table 6.1 shows some figures of our knowledge base, with over 31 million triples.

Table 6.1: Knowledge Base Figures

entity	number
classes	92
object properties	41
datatype property	12
named individuals	282
movies	447,451
TV series	60,758
Production/Company	131,646
Costume Designers	24,492
Actresses	522,027
Actors	592,692
Directors	165,745
Writers	239,619
Producers	272,600
Editors	97,088
Total triples	**31,728,919**

The dataset[1] is composed of 150 questions, divided into six different question types, which may include temporal attributes, sorting criteria, and grouping. Table 6.2 shows the distribution of questions by type.

Table 6.2: Distribution of Questions by Type

	what	who	count	when	yes/no	where
Number	94	25	11	9	6	5

Tables 6.3, 6.4, and 6.5 show the distribution of questions that include temporal attributes, sorting, and grouping criteria, respectively.

Table 6.3: Question filtering with time filter

	temporal filter	no temporal filter
Number	16	134

Table 6.4: Distribution of questions with sorting criteria

	sorting	no sorting
Number	24	126

Table 6.5: Distribution of questions with grouping criteria

	aggregation	no aggregation
Number	19	131

6.2 Question Interpreter

From this knowledge base, we apply our approach to each of the questions and evaluate the result obtained through the *Precision* and *Recall* metrics (aggregating them into the *F-score* metric). Table 6.6 shows the mean and variance of each metric.

[1]The dataset of questions is available in a repository on GitHub: https://github.com/alyssongomes/dataset-questions-imdb.

Table 6.6: Results averages

	Precision	Recall	F-score
mean	0.58	0.62	0.57
variance	0.239	0.231	0.231

In the table above, we can see that, on average, Recall is higher than Precision, and this indicates that most of the relevant answers were selected. However, part of what was selected is not relevant, evidenced by Precision. This shows a difficulty in correctly filtering the triples.

Table 6.7 shows the mean and variance of the F-score result broken down by each question type in the dataset. Here we can see that the best results came from questions where a location attribute (*where*) or a temporal attribute (*when*) were requested. This occurs because, in both cases, the search space is smaller due to the reduced number of properties and classes associated with geographic or temporal entities.

Table 6.7: F-score mean and variance for each question type

	count	what	when	where	who	yes/no
mean	0.45	0.53	0.77	1.00	0.76	0.00
variance	0.27	0.23	0.19	0.00	0.19	0.00

In contrast, the worst result is concentrated on *yes/no* questions, that is, questions that evaluate the existence of a particular set of triples. This happens because this type of question is very similar to *what* questions, which makes them very difficult to distinguish.

Tables 6.8, 6.9, and 6.10 show the mean and variance of the F-score, broken down by the filtering criteria types: temporal, aggregation, and sorting respectively.

Table 6.8: F-score mean and variance by time filter questions

	temporal filter	no temporal filter
mean	0.37	0.59
variance	0.25	0.23

Table 6.9: F-score mean and variance by questions with aggregation criteria

	aggregation	no aggregation
mean	0.46	0.58
variance	0.23	0.23

Table 6.10: F-score mean and variance by questions with sorting criteria

	sort	no sort
mean	0.54	0.57
variance	0.25	0.23

The greatest difficulty in applying these operations is choosing the criteria, as they can be referenced in a variety of ways, for example: *Which movie is the longest?* In this case, we must apply a sort operation to the *runtime* (duration) property that is associated with the movie entity (*Movie*). In another case, such as: *Which 5 actors acted most often in movies?*, it is necessary to apply the ordering operation to the number of films in which the actors acted.

These subtleties between the criteria are not always explicit in the question, which makes it difficult to determine the criterion to be used.

6.2.1 Comparing with a Baseline

To make a direct comparison, we built a *baseline* based on *templates*. These templates were developed specifically for each type of question, and for example, Listing 6.1 shows an example template defined for questions that ask for some temporal property.

To fill in the corresponding *slots*, we use the identified entities (actors, movies, TV series) in the knowledge base and the properties mentioned in the questions, identified using the ontology. To choose the most appropriate question type template, we use the factual category keywords presented in

Table 4.1.

```
prefix mo: <http://www.movieontology.org/2009/10/01/movieontology.owl#>
prefix dbo: <http://dbpedia.org/ontology/>
select <slot_value_property> where {
    <slot_individual1> a <slot_class>.
    <slot_individual1> <slot_property> <slot_value_property>.
    <slot_individual1> ?property ?when.
    filter(?property in (mo:releasedate, mo:indicationDate, dbo:birthDate))
}
```

Listing 6.1: Example of Template

As with the proposed methodology, we applied this baseline to the questions and obtained the results shown in Table 6.11.

Table 6.11: Averages and variance of baseline results in comparison to our approach

		Precision	Recall	F-score
baseline	mean	0.41	**0.64**	0.42
	variance	0.22	0.22	0.21
our approach	mean	**0.58**	0.62	**0.57**
	variance	0.239	0.23	0.23

In Table 6.11, we can see that Recall is higher than our approach, but this is due to the static structure of the queries, which causes more results to be returned, increasing the chances of the correct answer being returned even though there may be too many wrong answers, causing Precision to decrease.

Table 6.12 shows the baseline results for each question type.

Table 6.12: F-score mean and variance for each baseline question type in comparison to our approach

		count	what	when	where	who	yes/no
baseline	mean	0.27	0.42	0.60	0.60	0.47	0.00
	variance	0.21	0.23	0.18	0.30	0.18	0.00
our approach	mean	**0.45**	**0.53**	**0.77**	**1.00**	**0.76**	0.00
	variance	0.27	0.23	0.19	0.00	0.19	0.00

From the results of each question type, we can notice the impact of the static structure of the templates because the best results are concentrated in the questions of type *when* and *where*, where there is less variation than other types. In contrast, in all other types of questions there is a considerable drop in results.

Compared to our approach, where queries are dynamically generated, we have a significantly higher precision, because queries have better filters, eliminating irrelevant results. This can also be seen in questions where there is greater variety, such as questions of type *count* and *what*.

Tables 6.13, 6.14, and 6.15 show the mean and variance of the baseline F-score, broken down by the filter criteria types: temporal, aggregation and ordering respectively.

Table 6.13: Baseline f-score mean and variance by time-filter questions in comparison to our approach

		temporal filter	no temporal filter
baseline	mean	0.05	0.46
	variance	0.03	0.22
our approach	mean	**0.37**	**0.59**
	variance	0.25	0.23

Table 6.14: Baseline f-score mean and variance by questions with aggregation criteria in comparison to our approach

		aggregation	no aggregation
baseline	mean	0.19	0.45
	variance	0.12	0.22
our approach	mean	**0.46**	**0.58**
	variance	0.23	0.23

Table 6.15: Baseline f-score mean and variance by questions with sorting criteria in comparison to our approach

		sort	no sort
baseline	mean	0.16	0.47
	variance	0.10	0.22
our approach	mean	**0.54**	**0.57**
	variance	0.25	0.23

Finally, the tables above show baseline results for questions that require specific operations such as aggregations and sorting. Moreover, in these templates, the problem of determining the correct arguments to pass to aggregate or sort operators remains and is maximized in many cases.

Compared to our approach, dependency trees are more effective in determining these parameters because their structure gives better indications of which parameters to use, resulting in better results.

6.2.2 Comparing with other works

As a benchmark for our approach, we took the winning work of the QALD 9 competition, which we took as a basis for building our dataset. It is important to note that we will not be able to make a direct comparison because the question dataset and knowledge base are different, and the knowledge base used in the competition is dynamic, as new data are periodically entered and the competition was held in 2018. Also, we do not have a canonical question dataset to use as an evaluation set for this task, as many domain-specific works produce their own datasets, which motivated our effort to develop this question dataset.

However, question sets are similar in terms of question variety, in that they cover the most common types of knowledge base questions, as shown in Table 6.16.

Table 6.16: Distribution of questions by type of the dataset of QALD

	what	who	count	when	yes/no	where
Number	84	26	19	11	5	5

Table 6.17 shows the ranking of the competition we are considering. From the results, it is possible to notice the degree of difficulty of the task, especially considering the domain amplitude.

Work	Authors	Precision	Recall	F-score
gAnswer	**Hu et al. (2018)**	**0.293**	**0.327**	**0.298**
wdaqua-core-1	Diefenbach et al. (2018)	0.261	0.267	0.250
TeBaQA	Nancke et al.	0.129	0.134	0.130
QASystem	Bence et al.	0.097	0.116	0.098
Elon	Blübaum and Düsterhus	0.049	0.053	0.050

Table 6.17: Results of the QALD 9

Let us consider the winner of the competition, Hu et al. (2018). Their work also uses a graph-oriented paradigm to structure the search triples. However, this graph is used to structure the question given as input and other resources. Machine learning models (BiLSTM-CRF) for entity identification, entity linking algorithms for extracting semantic relationships, along with dependency trees and co-reference are used. However, this approach was developed for the English language.

It is possible to notice similarities between the winner of the competition and the present work, since the results give a strong indication that the graph-oriented paradigm and the dependency trees are promising approaches, even when applied in different languages and domains.

6.2.3 Performance Analysis

About the time of execution of queries, Tables 6.18 and 6.19 show the mean, standard deviation, and variance of time of execution in seconds.

Table 6.18: Time of execution of queries in seconds

Mean	Standard Deviation	Variance
1.33	7.15	51.20

Table 6.19: Time of execution of queries for each question type

	count	what	when	where	who	yes/no
mean	0.93	1.89	0.01	0.02	0.42	0.009
variance	3.13	79.75	0.00	0.00	1.03	0.00
standard deviation	1.77	8.93	0.00	0.02	1.01	0.00

In addition to the most recurring question types, the Tables 6.20, 6.21 and 6.22 the mean, standard deviation, and variance of each query type by considering the types of operations most commonly applied to queries, such as aggregation, sorting, and time filtering.

Table 6.20: Query execution time by question type and by temporal filter existence

	type question	mean	standard deviation	variance
no temporal filter	count	0.97	2.30	$5.31 * 10^{0}$
	what	1.93	8.80	$7.75 * 10^{1}$
	when	0.00	0.00	$5.63 * 10^{-7}$
	where	0.01	0.00	$7.74 * 10^{-5}$
	who	0.32	0.74	$5.61 * 10^{-1}$
	yes/no	0.00	0.00	$4.87 * 10^{-7}$
temporal filter	count	**2.06**	0.00	0.00
	what	1.39	3.56	$1.27 * 10^{1}$
	when	0.02	0.02	$5.93 * 10^{-4}$
	yes/no	0.00	0.00	0.00

Table 6.21: Query execution time by question type and by aggregate existence

	type question	mean	standard deviation	variance
no aggregation	count	1.24	2.58	$6.70 * 10^{0}$
	what	1.70	8.64	$7.47 * 10^{1}$
	when	0.01	0.01	$1.22 * 10^{-1}$
	where	0.01	0.00	$7.74 * 10^{-5}$
	who	0.25	0.66	$4.39 * 10^{-1}$
	yes/no	0.00	0.00	$3.97 * 10^{-7}$
aggregation	count	0.707	1.17	$1.38 * 10^{0}$
	what	**2.89**	**5.55**	$3.08 * 10^{0}$
	who	2.12	0.00	0.00

Table 6.22: Query execution time by question type and by sort existence

	type question	mean	standard deviation	variance
no sort	count	1.20	2.30	$5.29 * 10^{0}$
	what	1.38	8.87	$7.87 * 10^{1}$
	when	0.01	0.01	$1.22 * 10^{-4}$
	where	0.01	0.00	$7.74 * 10^{-5}$
	who	0.17	0.55	$3.02 * 10^{-1}$
	yes/no	0.00	0.00	$3.97 * 10^{-7}$
sort	count	0.01	0.00	0.00
	what	**3.47**	**5.85**	$3.42 * 10^{1}$
	who	2.10	0.04	$1.70 * 10^{-3}$

As we highlighted in the tables above, what type *what* questions that use some operation (grouping, sorting, or time filtering) are the questions that require the most processing time, because this type of question requires listing a set of entities that satisfy a more complex set of constraints that require equally complex operations. We can notice this complexity in the Tables 6.21 and 6.22 because they are the largest means and standard deviations.

For example, Figure 6.1 shows two examples of costly queries. In the example *a*, it is necessary to group the actors from the films they performed. Given the large volume of movies and actors, this becomes the most expensive query. In the example *b*, the large volume of movies also impacts query response time, as there is also a need to apply a filter to the location of the movie.

Quais os 5 atores que atuaram mais vezes em filmes?

```
SELECT ?birthName (COUNT(*) as ?count)
WHERE {
        ?m a :Movie .
        ?a a :Actor .
        ?m :hasMaleActor ?a .
        ?a :birthName ?birthName .
}
GROUP BY ?birthName
ORDER BY DESC(?count)
LIMIT 5
```

(a)

Quais os filmes europeus mais bem avaliados?

```
SELECT ?title ?imdbrating
WHERE {
        ?m a :Movie .
        ?t a :Europe .
        ?m :imdbrating ?imdbrating .
        ?m :hasFilmLocation ?t .
        ?m :title ?title .
}
ORDER BY DESC(?imdbrating)
```

(b)

Figure 6.1: Examples of expensive queries

6.2.4
Error Analysis

We analyzed the most frequent errors in the question sets. Table 6.23 shows the distribution of the questions that received incorrect answers.

Table 6.23: Distribution of question types that received incorrect answers

	what	who	count	yes/no	when
Total Questions	94	25	11	6	5
Wrong answers	36	6	6	6	2
Error Percentage	38.2%	24%	54%	100%	40%

As can be seen in Table 6.23, questions of type *what* concentrate most of the errors. This may happen due to the wide range of questions that can be made, in various ways. This factor strongly impacts the dependency tree generation, as its structure can vary considerably depending on the way the question is formulated, as exemplified in Figure 6.2. This variation affects the generation of query search triples, and the selection of the entities that will be used in grouping and sorting operations, increasing the chances of generating wrong queries, and consequently impacting also questions of type *count*.

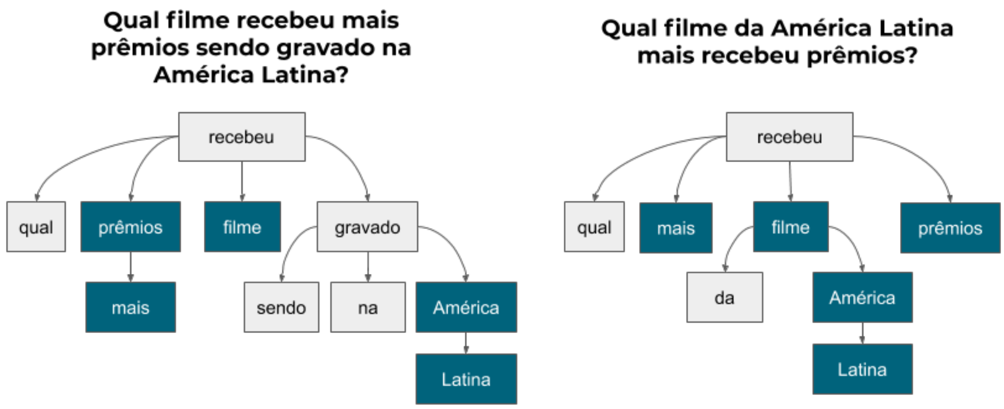

Figure 6.2: Example of variations in the dependency trees

As mentioned earlier, our approach still has difficulty distinguishing questions of the type *yes/no*, usually because they are very similar to questions of type *what*, as exemplified in Figure 6.3. However, the main challenge is to determine when only to test the existence of a particular set of triples and when it is necessary to return its results. Moreover, these cases were not mapped in our approach, but some of the data are missing, so both factors contribute to none of the questions being answered.

Johnny Depp participou de A Star Is Born?

```
ASK {
      :name/231595 :hasMaleActor :title/15153
}
```

Johnny Depp participou de quais filmes?

```
SELECT ?title
WHERE{
      ?m a :Movie.
      name/1004637 :hasMaleActor ?m.
      ?m :title ?title.
}
```

Figure 6.3: Example of variations between *what* and *yes/no* type questions

The errors associated with questions of the type *who* are strongly related to the inability to identify when it is necessary to return individuals of more than one class type, for example: in the question "Quem faz parte do elenco de Moonlight?", the term "elenco" (*cast*) means that both actors and actresses must be returned. However, our approach is not yet able to detect cases like this.

Finally, when dealing specifically with questions of the type *when*, the wrong answers are due to the difficulty of correctly determining the property that will be used in the query time filter, as they are not always mentioned explicitly in the question, as exemplified in Figure 6.4.

Quando o ator Robin Williams ganhou o Oscar?

```
SELECT ?date
WHERE{
      name/1004637 :isAwardedWith :Oscar_Award.
      name/1004637 :indicationDate ?date.
}
```

Quando Robin Williams nasceu?

```
SELECT ?date
WHERE{
      name/1004637 :birthDate ?date.
}
```

Figure 6.4: Example of variations in question of type *when*

Therefore, most errors are associated with variability in the formation of questions or cases not mapped in our approach. This indicates the need to map other patterns that may occur in the dependency tree; It is also necessary to add new cases that consider only mentions to individuals to interpret the *yes/no* questions, as well as seeking more contextual information to answer other types of questions better.

6.3 Generator-Related Questions

In order to avoid distortions in the evaluations, some related questions were removed from the experiment, due to the lack of meaning within the context of the main questions, for example: from the question "Which 5 films had the highest gross revenues?", our approach could eventually raise the following related question: "Which 5 Actresses Had the Highest Gross Revenue?". However, this question should be rephrased to the following format: "Which 5 Actresses Performed in Movies That Had the Highest Gross Revenue?".

These distortions result from a limitation of our generator, which does not yet correctly verbalize the path that links the primary entity to a semantically more distant entity, whose term cannot be simply replaced by another, as is the case of the previous example. The following paragraphs describe and discuss the results obtained when users evaluated the questions generated by our approach.

6.3.1 Analysis between participant groups

The forms containing the two question groups (P and A) were sent to two groups of participants, G1 and G2, which totaled 42 answers. Most of the participants are undergraduate and postgraduate students, divided into some areas such as Computer Science, Chemical Engineering, and Electronics, as shown in Tables 6.24 and 6.25.

Table 6.24: Distribution of participants per disciplinary background

Formation	Number
Incomplete Graduation	2
Graduate	8
Incomplete Post Graduate	12
Postgraduate	20

Table 6.25: Distribution of participants by area

Area	Number
Computation	37
Chemical engineering	1
Physical Education	1
Industrial	1
Electronic Engineering	1
Economy	1

We first look at the results about the related questions and note that in general, participants did indeed find the suggested questions to be related to the stated question. Figures 6.6, 6.7, and 6.8 show how the distribution of the degree to which each question was deemed useful in the groups of questions sorted in the proposed order. Conversely, Figures 6.10, 6.11, and 6.12 show the results for the groups of questions in arbitrary order.

In the figures below, we have sorted the questions according to the criteria proposed in this paper. In the distributions, it is possible to notice a tendency in the evaluations, because the number of evaluations that indicate the lack of relationship grows as the question has lower positions in the ranking.

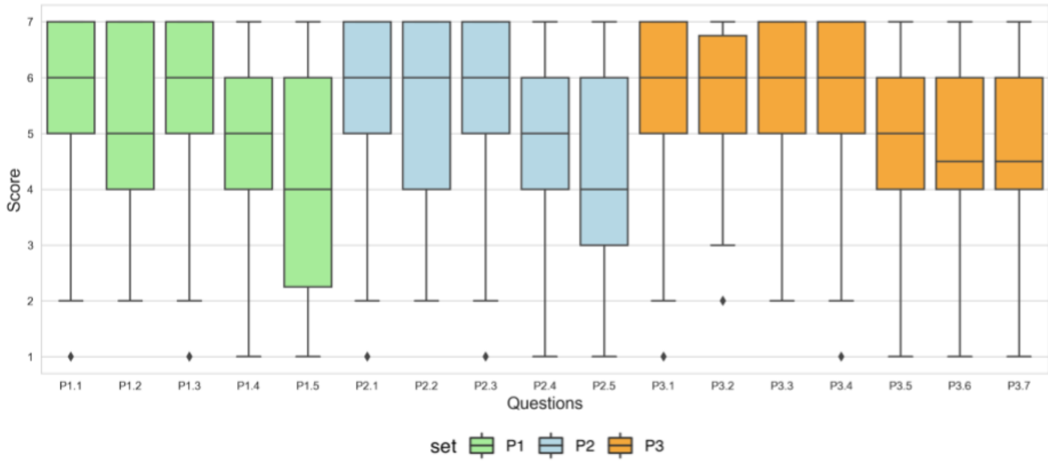

Figure 6.5: Compiled Distribution of Group P Assessments

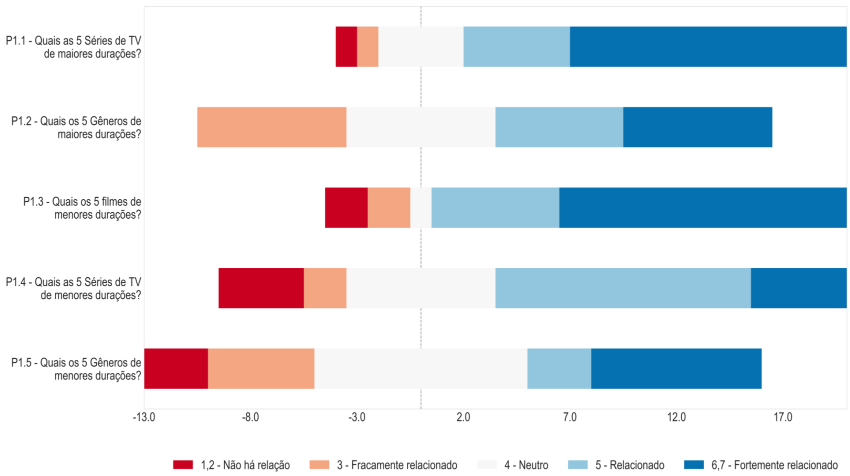

Figure 6.6: Distribution of assessments of related questions from group P1: *Quais os 5 filmes de maiores durações?*

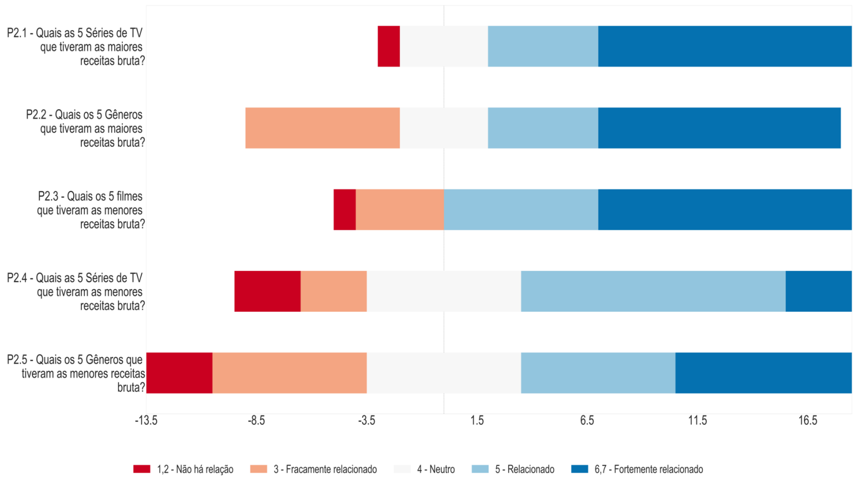

Figure 6.7: Distribution of evaluations of group P2 related questions: *Quais os 5 filmes que tiveram as maiores receitas bruta?*

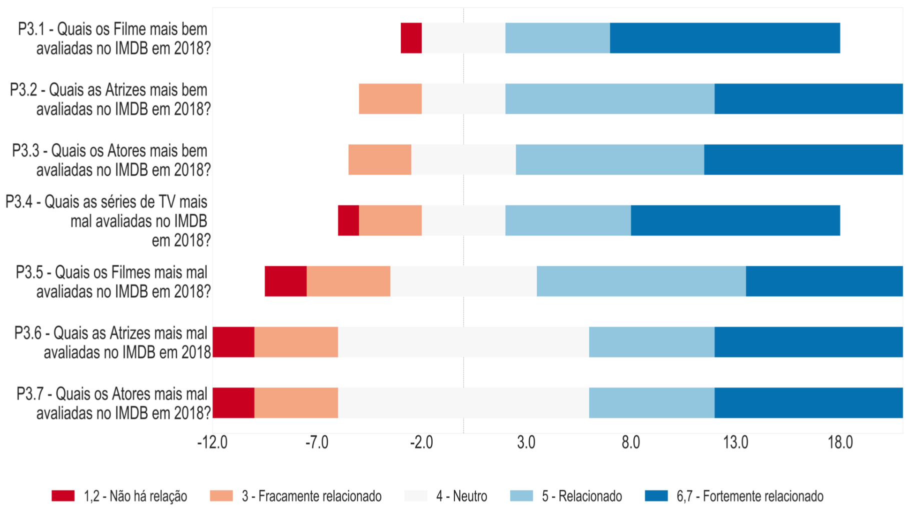

Figure 6.8: Distribution of ratings for related questions from group P3: *Quais as séries de TV mais bem avaliadas no IMDB em 2018?*

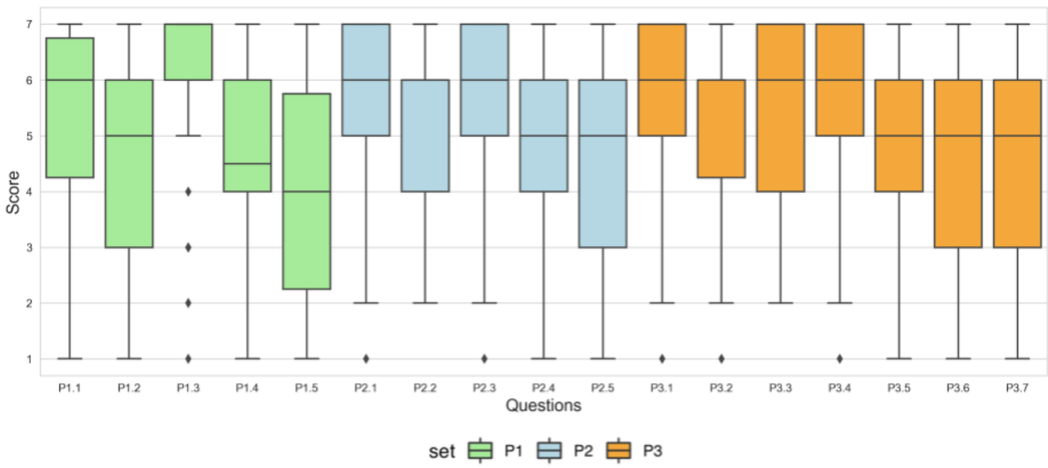

Figure 6.9: Compiled Distribution of Group A Assessments

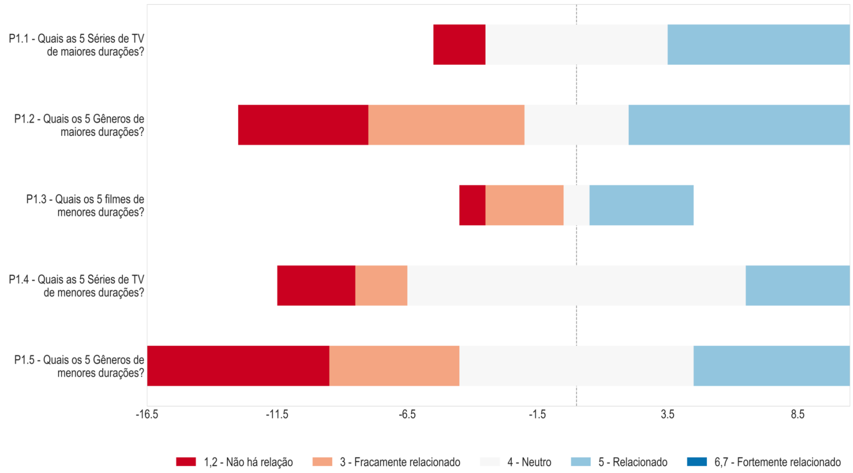

Figure 6.10: Distribution of evaluations of related questions from group A1: *Quais os 5 filmes de maiores durações?*

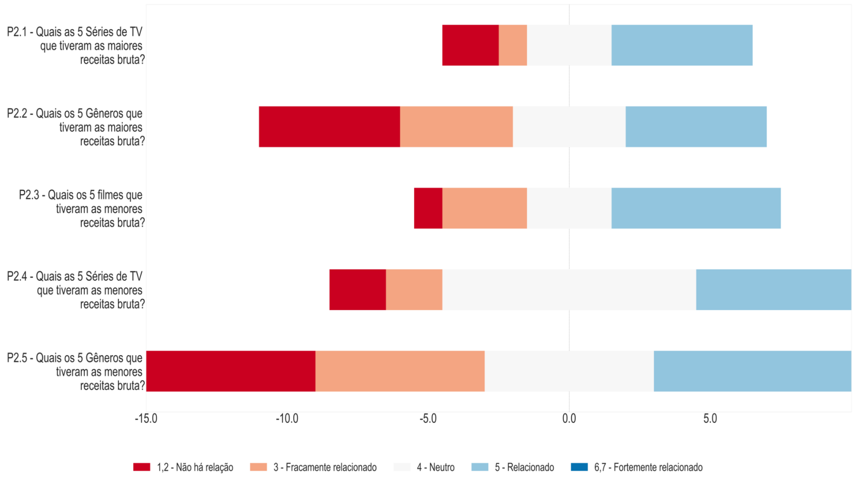

Figure 6.11: Distribution of evaluations of related questions from group A2: *Quais os 5 filmes que tiveram as maiores receitas bruta?*

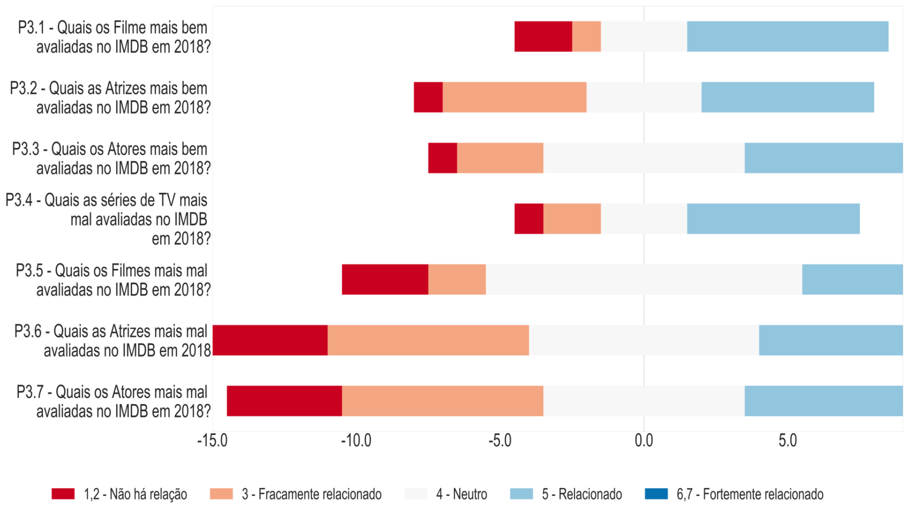

Figure 6.12: Distribution of evaluations of related questions from group A3: *Quais as séries de TV mais bem avaliadas no IMDB em 2018?*

In addition to individual questions, participants also assessed the adequacy of each set (P1, P2, P3, A1, A2, and A3) and the order of questions within each set, considering how related its questions were to the initial question in each group.

So, we first check whether the question groups have a significant difference between the groups of participants (G1 and G2), that is, whether the score assigned to the order/set of P1 of G1 is significantly different from P1 of G2, similarly in the P2, P3, A1, A2 and A3 from both groups. For this, we applied the Mann-Whitney test in the question groups of each group of participants. Considering $\alpha = 0.05$, we found that there was no significant difference in any of them, as shown in Tables 6.26 and 6.27.

Table 6.26: Mann-Whitney results on the scores assigned to each set of questions, against each participant group

Group of Question	Mann-Whitney U	p-value
P1	195.5	0.5255927
P2	185.0	0.3699915
P3	146.0	0.0574410
A1	221.5	0.9795426
A2	213.0	0.8664247
A3	175.5	0.2589675

Table 6.27: Mann-Whitney results on the scores assigned to the order of questions of each set, against each participant group

Group of Question	Mann-Whitney U	p-value
P1	208.0	0.7688985
P2	176.5	0.2748184
P3	157.0	0.1084168
A1	227.0	0.8676748
A2	209.5	0.7983183
A3	193.0	0.4996741

Because there was no significant difference between order and set evaluations between participant groups, we can consider that the order in which the question groups were listed was not a significant factor in the evaluation. Therefore, the subsequent analyses will be made joining the respective question groups, *i.e.*, the scores of P1 of G1 and P1 of G2 will be joined into a single group P1, and the same will occur for the other groups.

6.3.2 Comparing the proposed ranking and the random ranking

After analyzing the question groups separately, we will analyze the evaluations of the question group in the proposed order (P) against the arbitrary-order group (A).

First we look at the ratings given to the question groups as a whole for P and A. So we apply the Mann-Whitney test for each question group pair (P_i and A_i), and we found no significant difference between the groups, as shown in Table 6.28. This result was expected, as the groups are made up of the same sets of questions.

Table 6.28: Mann-Whitney results on the scores assigned to each set of questions against each ranking

Pair of group	Mann-Whitney U	p-value
P1, A1	0.300424	0.5836165
P2, A2	0.2508909	0.6164485
P3, A3	0.2383102	0.6254293

We then performed the same assessment on the scores assigned to the order of each group of P and A. For this, we also applied the Mann-Whitney test, and found a significant difference between P3 and A3, as shown in Table 6.29.

Table 6.29: Mann-Whitney results on the scores assigned to the order of questions in each group

Pair of group	Mann-Whitney U	p-value
P1, A1	973.0	0.4108719
P2, A2	1041.0	0.1525532
P3, A3	**1204.5**	**0.0035341**

A difference in the assessment of the order of Pi and A1 was expected, as the order is the only distinguishing feature of question groups. However, the fact that there is a significant difference only between P3 and A3 reveals that the effect of the order was only noticeable in the larger group, as P3 and A3 had 7 related questions, while the other groups had only 5.

Table 6.30 shows the mean and median score of each group of questions, highlighting those with the greatest difference between the metrics. In Figure 6.13 we graphically show the results from Table 6.30.

Table 6.30: Statistics of the scores assigned to the order of questions of each set, for each ranking

Group	Mean	Median
P1	6.761905	7.0
A1	6.261905	7.0
P2	6.47619	7.0
A2	5.666667	6.0
P3	**7.547619**	**8.0**
A3	**6.142857**	**6.5**

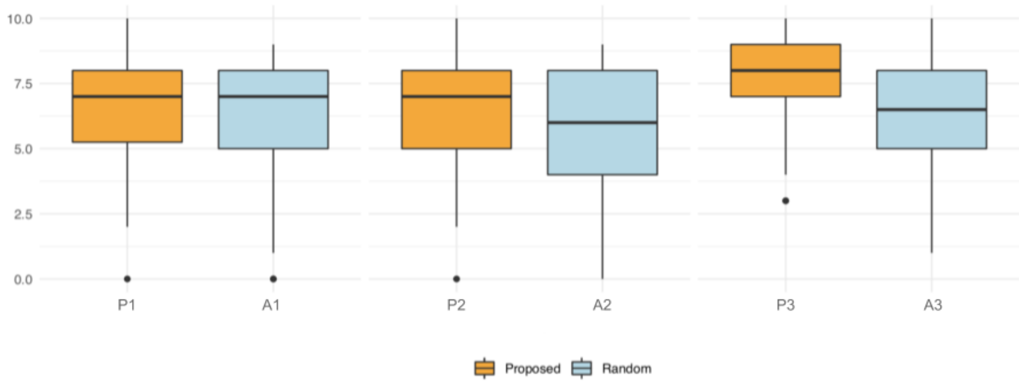

Figure 6.13: Distribution of grades assigned to order by question group

6.3.3 Evaluation of the effect of ordering

In addition to assessing the impact of ordering on the question set, we also analyze the impact of ordering on the scores that were given to the individual questions.

First, we check whether there was a significant difference between the questions of each ordered group P1, P2, and P3. For this, we applied the Kruskal-Wallis test. Considering $\alpha = 0.05$, we found significant differences, as shown in Table 6.31.

Table 6.31: Kruskal-Wallis results on the scores assigned to each question in a set.

Group	Kruskal-Wallis $\mathcal{X}^2$	p-value
P1	**26.60436**	**0.0000239**
P2	**24.79042**	**0.0000554**
P3	**26.00734**	**0.0002219**

Knowing that there is a significant difference between the question groups, it is necessary to verify where precisely these differences occur, so we applied the Conover-Iman post-hoc test to the question pairs in each group, whose results are presented in Tables 6.32, 6.33, and 6.34.

Table 6.32: Results of Conover-Iman post-hoc test for questions in P1

pair	$\mathcal{X}^2$	adjusted p-value
P1.1 - P1.2	26.60436	0.5469912
P1.1 - P1.3	26.60436	1.0000000
P1.2 - P1.3	26.60436	0.1670326
P1.1 - P1.4	**26.60436**	**0.0112686**
P1.2 - P1.4	26.60436	0.6945731
P1.3 - P1.4	**26.60436**	**0.0018080**
P1.1 - P1.5	**26.60436**	**0.0004995**
P1.2 - P1.5	26.60436	0.0958460
P1.3 - P1.5	**26.60436**	**0.0000563**
P1.4 - P1.5	26.60436	1.0000000

Table 6.33: Results of Conover-Iman post-hoc test for questions in P2

pair	$\mathcal{X}^2$	adjusted p-value
P2.1 - P2.2	24.79042	1.0000000
P2.1 - P2.3	24.79042	1.0000000
P2.2 - P2.3	24.79042	1.0000000
P2.1 - P2.4	**24.79042**	**0.0185959**
P2.2 - P2.4	24.79042	0.1421764
P2.3 - P2.4	**24.79042**	**0.0073375**
P2.1 - P2.5	**24.79042**	**0.0006855**
P2.2 - P2.5	**24.79042**	**0.0091126**
P2.3 - P2.5	**24.79042**	**0.0002186**
P2.4 - P2.5	24.79042	1.0000000

Table 6.34: Results of Conover-Iman post-hoc test for questions in P3

pair	$\mathcal{X}^2$	adjusted p-value
P3.1 - P3.2	26.00734	1.0000000
P3.1 - P3.3	26.00734	1.0000000
P3.2 - P3.3	26.00734	1.0000000
P3.1 - P3.4	26.00734	1.0000000
P3.2 - P3.4	26.00734	1.0000000
P3.3 - P3.4	26.00734	1.0000000
P3.1 - P3.5	**26.00734**	**0.0099449**
P3.2 - P3.5	26.00734	0.8290667
P3.3 - P3.5	26.00734	0.7727853
P3.4 - P3.5	26.00734	0.1992686
P3.1 - P3.6	**26.00734**	**0.0014447**
P3.2 - P3.6	26.00734	0.2405417
P3.3 - P3.6	26.00734	0.2210074
P3.4 - P3.6	**26.00734**	**0.0444130**
P3.5 - P3.6	26.00734	1.0000000
P3.1 - P3.7	**26.00734**	**0.0014447**
P3.2 - P3.7	26.00734	0.2405417
P3.3 - P3.7	26.00734	0.2210074
P3.4 - P3.7	**26.00734**	**0.0444130**
P3.5 - P3.7	26.00734	1.0000000
P3.6 - P3.7	26.00734	1.0000000

From these results, we note that there was a significant difference in some cases, which can be explained from the principle that guides our sorting criterion. This principle starts from the notion that the larger the number of entities modified from the initial question, the further we get from it and the recommended question is perceived as less related. From this principle, our ranking criterion applies a penalty to questions that modify more entities, lowering their ranking score.

Analyzing all cases where there was a significant difference, we can note that more than one entity was modified, so the question suffered a ranking penalty. The pairs of questions that differed always start from a question that has not been penalized to a question that has been penalized. This gives us clues that questions with more modified entities really should have lower priority, even if the strengths of relationships get other values, as exemplified in Table 6.35, where the first column shows the initial question with the main entities highlighted in bold, and the second column shows the related questions with the replaced entities also highlighted in bold.

Table 6.35: Examples of analyzed questions

Initial Question	Related Questions
P1 - Quais os 5 **filmes** de **maiores** durações?	P1.1 - Quais as 5 **Séries de TV** de maiores durações? P1.4 - Quais as 5 **Séries de TV** de **menores** durações?
P2 - Quais os 5 **filmes** que tiveram as **maiores** receitas bruta?	P2.1 - Quais as 5 **Séries de TV** que tiveram as maiores receitas bruta? P2.4 - Quais as 5 **Séries de TV** que tiveram as **menores** receitas bruta?
P3 - Quais as **séries de TV** mais **bem** avaliadas no IMDB em 2018?	P3.1 - Quais os **Filme** mais bem avaliadas no IMDB em 2018? P3.5 - Quais os **Filmes** mais **mal** avaliadas no IMDB em 2018?

These differences can also be noted if we compare the evaluation of the individual questions (Figures 6.5 and 6.9), considering the pairs of questions with a significant difference.

6.3.4 Review of participants' comments

In addition to the evaluations, at the end of the forms there was an open text field in which participants could add comments and suggestions about the evaluations. In all, 13 comments were added, which can be found in Appendix D.

The first type of recurring comment was the suggestion of new forms of ordering. In many cases, participants were able to identify a pattern in the list of questions, so they suggested incorporating other aspects into the sorting criteria, such as approaching questions that have opposite elements (such as “Longer Length Films” and “Shorter Length Films”, for example); other comments also corroborated our hypothesis that entities that were more distant in the ontology should also be in lower positions in the ranking.

Another type of recurring comment was the objection to entities we considered related in our experiment. Some participants reported that they did not consider certain entities as related (such as Movies and TV Series);

in contrast, other participants reported that they considered them as related entities, but that they could be sorted in some other way. This reveals the subjectivity of the concept of relationship, as each participant has a criterion for considering two entities as related. This may explain the reports of participants who said they did not understand what we were considering as related entities.

It is noteworthy that we do not consult the opinion of domain experts in the process of defining interesting relationships between the entities. On the one hand, their expert knowledge can make the relations more precise. On the other hand, their judgement may depart from the layperson's perspective.

In any case, each point raised in the participants' comments provides us with input to refine our approach in future work.

7
Conclusion

In this chapter, we present an overview of the main points of this dissertation and list works that can be developed in the future.

7.1
Main contributions

In this dissertation, we developed an approach that can answer questions in the natural language, specifically in Portuguese, by converting the questions into SPARQL queries, by using an ontology that describes a particular domain and a corresponding knowledge base. From this approach, we also developed a knowledge-discovery engine from the automatic generation of related questions.

Our approach was based on the works of Paredes-Valverde et al. (2015) and Li and Xu (2016), from which we take dependency trees as our primary tool, as a method to structure the input question, along with a graph-oriented paradigm applied to ontology, to find semantic relations between the entities mentioned in the question, allowing us to construct more elaborate queries that use the relationship between the cited terms, and not just use them independently as disconnected keywords.

To evaluate our approach, we built a question dataset about the movie domain (IMDB) database based on the QALD 9 (based on the DBpedia knowledge base) competition question dataset, consisting of 150 questions belonging to 6 different categories. We applied our approach to this dataset and obtained a Precision average of about 58% and a Recall average of 62% – an F-score average of 57%, with an average F-score above 50% on 4 of the 6 question types.

To evaluate our methodology for automatically generating related questions, we applied an online questionnaire. In total, 42 people evaluated the related questions generated by our approach and the order in which they were presented. These assessments provided us with promising results regarding how we produced related questions. In particular, we found statistically significant differences in favor of our ranking criterion for related questions.

7.2 Future Work

To support this section of future work, we have identified some limitations that will guide later works.

The first of these is the need to calibrate the weights of the graph of the ontology manually. This calibration is necessary because specific paths in ontology are preferable than others, depending on the domain in which our approach is applied. We believe an automated way of calibrating these weights may improve the process.

A possible solution is the methodology used in the training process of neural network learning algorithms: a question dataset would be given as input to the model, whose answers would be given as input to an error function, whose result in turn would serve as input to update the graph weights.

One limitation we identified in generating related questions was the verbalization of questions that are semantically more distant from the original question. For example, a user might ask a question like “Which movies had the longest duration?” but, in the relationship definition phase, it was defined that the *Movie* entity is related to the *Actor*. Then based on this annotation we could replace the term *Movie* present in the question with the term *Actor*. However, this question would be meaningless, as the transition between entities must take into account the context of question.

A possible solution to this problem would be to consider the path that links the two entities in the ontology. Therefore, in the question “Which films had the longest duration?”, assuming that there is a property named *acted on*, which links the *Movie* and *Actor* entities, we will consider this property within the process, so our related question would be “Which actors played in films that had the longest durations?”.

Another future work that would assist in the process of generating related questions is the development of a tool that would support the user in the process of defining relationships of interest. However, even with the support of some tool that allows manipulating ontologies, this process depends on previous knowledge about how the ontology is structured.

A graphical tool that showed the ontology and allowed to define relationships and the strengths of these relationships in an understandable way would greatly aid the definition process.

To expand the range of related information provided by our approach, we also propose as future work the use of *Serendipity Patterns*. Serendipity patterns are the various ways of finding a particular piece of information or knowledge that was not initially sought, that is, found occasionally. In future work, these standards can be adapted and integrated into our approach using the methodology presented in the work of Jeronimo (2018).

To improve the ranking quality of related questions, we also propose as future work to adapt and integrate into our approach the strategy defined in the work of Menendez (2019).

Their work has formalized a set of principles for defining a family of measures of importance to nodes in RDF graphs. Applying this information to the data of a given domain, we can obtain the classes of the entities present in the ranking and rank the related questions according to the order of the classes obtained.

Finally, we intend to apply our approach to other domains, such as musical, educational, and so on, to ascertain the generalizability of our method, provided there is an ontology that describes the respective domain.

Bibliography

14224:2016, I. (2016). *ISO 14224:2016 (OGP) Petroleum, petrochemical and natural gas industries – Collection and exchange of reliability and maintenance data for equipment*. ISO.

Baudiš, P. and Šedivỳ, J. (2015). Modeling of the question answering task in the yodaqa system. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 222–228. Springer.

Berant, J. and Liang, P. (2014). Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1415–1425.

Berners-Lee, T. (1994). W3 future directions. *Plenary at WWW Geneva 94*.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific american*, 284(5):34–43.

Capindale, R. A. and Crawford, R. G. (1990). Using a natural language interface with casual users. *International Journal of Man-Machine Studies*, 32(3):341–361.

Cortes, E. G., Woloszyn, V., and Barone, D. A. (2018). When, where, who, what or why? a hybrid model to question answering systems. In *International Conference on Computational Processing of the Portuguese Language*, pages 136–146. Springer.

Damljanovic, D., Agatonovic, M., and Cunningham, H. (2012). Freya: An interactive way of querying linked data using natural language. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

De Marneffe, M.-C. and Manning, C. D. (2008). The stanford typed dependencies representation. In *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pages 1–8. Association for Computational Linguistics.

Diefenbach, D., Singh, K., and Maret, P. (2018). On the scalability of the qa system wdaqua-core1.

Dubey, M., Dasgupta, S., Sharma, A., Höffner, K., and Lehmann, J. (2016). Asknow: A framework for natural language query formalization in sparql. In *International Semantic Web Conference*, pages 300–316. Springer.

Gao, T., Dontcheva, M., Adar, E., Liu, Z., and Karahalios, K. G. (2015). Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 489–500. ACM.

Gonçalo Oliveira, H. and Gomes, P. (2014). ECO and Onto.PT: A flexible approach for creating a Portuguese wordnet automatically. *Language Resources and Evaluation*, 48(2):373–393.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

Hakimov, S., Tunc, H., Akimaliev, M., and Dogdu, E. (2013). Semantic question answering system over linked data using relational patterns. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 83–88. ACM.

Harris, S., Seaborne, A., and Prud'hommeaux, E. (2013). Sparql 1.1 query language. *W3C recommendation*, 21(10):778.

Hartawan, A., Suhartono, D., et al. (2015). Using vector space model in question answering system. *Procedia Computer Science*, 59:305–311.

Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., and Ngonga Ngomo, A. C. (2017). Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6):895–920.

Hu, S., Zou, L., and Zhang, X. (2018). A state-transition framework to answer complex questions over knowledge base. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2098–2108.

Jeronimo, S. d. A. E. (2018). *Exploring RDF Knowledge Bases through Serendipity Patterns*. PhD thesis, Pontifical Catholic University of Rio de Janeiro.

Jurafsky, D. and Martin, J. H. (2014). *Speech and language processing*, volume 3. Pearson London.

Kao, A. and Poteet, S. R. (2007). *Natural language processing and text mining.*

Kaufmann, E. and Bernstein, A. (2007). How useful are natural language interfaces to the semantic web for casual end-users? In *The Semantic Web*, pages 281–294. Springer.

Kaufmann, E., Bernstein, A., and Zumstein, R. (2006). Querix: A natural language interface to query ontologies based on clarification dialogs. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, pages 980–981.

Lassila, O., Swick, R. R., Wide, W., and Consortium, W. (1998). Resource description framework (RDF) model and syntax specification.

Li, H. and Xu, F. (2016). Question answering with dbpedia based on the dependency parser and entity-centric index. In *Computational Intelligence and Applications (ICCIA), 2016 International Conference on*, pages 41–45. IEEE.

Lopez, V., Tommasi, P., Kotoulas, S., and Wu, J. (2016). Queriodali: question answering over dynamic and linked knowledge graphs. In *International Semantic Web Conference*, pages 363–382. Springer.

Manning, C. D., Manning, C. D., and Schütze, H. (1999). *Foundations of statistical natural language processing.* MIT press.

Marchionini, G. (1997). *Information seeking in electronic environments.* Number 9. Cambridge university press.

McGuinness, D. L., Van Harmelen, F., et al. (2004). Owl web ontology language overview. *W3C recommendation*, 10(10):2004.

Melo, D., Rodrigues, I. P., and Nogueira, V. B. (2016). Using a dialogue manager to improve semantic web search. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 12(1):62–78.

Menendez, E. S. (2019). Novel node importance measures to improve keyword search over rdf graphs.

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Paredes-Valverde, M. A., Rodríguez-García, M. Á., Ruiz-Martínez, A., Valencia-García, R., and Alor-Hernández, G. (2015). Onli: an ontology-based system for querying dbpedia using natural language paradigm. *Expert Systems with Applications*, 42(12):5163–5176.

Reinsel, D., Gantz, J., and Rydning, J. (2018). The digitization of the world: from edge to core. *IDC White Paper Doc# US44413318. Viewed March.*

Rodrigues, R. and Gomes, P. (2015). Rapport—a portuguese question-answering system. In *Portuguese Conference on Artificial Intelligence*, pages 771–782. Springer.

Setlur, V., Battersby, S. E., Tory, M., Gossweiler, R., and Chang, A. X. (2016). Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 365–377. ACM.

Sowa, J. F. (1994). *Knowledge representation: logical, philosophical, and computational foundations*, volume 9. Brooks/Cole.

Sun, Y., Leigh, J., Johnson, A., and Lee, S. (2010). Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *International Symposium on Smart Graphics*, pages 184–195. Springer.

Thanawala, P., Pareek, J., and Shah, M. (2014). Ontobaeval: Ontology based automatic evaluation of free-text response. In *Technology for Education (T4E), 2014 IEEE Sixth International Conference on*, pages 189–190. IEEE.

Toti, D. (2014). Aqueos: A system for question answering over semantic data. In *Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on*, pages 716–719. IEEE.

Yang, M.-C., Lee, D.-G., Park, S.-Y., and Rim, H.-C. (2015). Knowledge-based question answering using the semantic embedding space. *Expert Systems with Applications*, 42(23):9086–9104.

Yao, X. and Van Durme, B. (2014). Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 956–966.

A
Algorithms

In this appendix, we present the two main algorithms used in this work. Algorithm 1 is responsible for converting the path between two entities into a set of query triplets that are used in the SPARQL query. This method uses a main data structure (*triples*) where it stores the definitions of entities and their relationships.

Algorithm 2 is responsible for navigating the dependency tree, first evaluating the root vertex, and then the subtree nodes. As mentions of classes or individuals are found, algorithm 1 is used to convert the path that joins these entities into query triplets.

Algorithm 1 Transformation algorithm from graph path into query triplets

```
triples ← {}                                              ▷ key-value structure
LABEL ← edge labels
REF ← reference of the last vertex representing an entity
function DEFINE_REFERENCE(uri)
    var ← define a variable that will only represent uri within triples_query
if it is already in triples_query
return var
end function

function RELATION_TO_QUERY_TUPLES(path)
    for i ∈ |path| do
        source, target ← path[i][SOURCE], path[i][TARGET]
        N_source ← path[i + 1][SOURCE]
        N_target ← path[i + 1][TARGET]
        s ← DEFINE_REFERENCE(source)
        t ← DEFINE_REFERENCE(target)
        if 'inverse' ∈ LABEL[source, target] then
            if REF[source, target] then
                add ('a',s) in triples[t]
                (LABEL[source, target], REF[source, target]) in triples[t]
            else
                add ('a',s) and (LABEL[source, target],s) in triples[t]
            end if
            REF[N_source, N_target] ← t
        else if 'subClassOf' ∈ LABEL[source, target] OR 'classFatherOf' ∈
LABEL[source, target] then
            if not REF[source, target] then
                add ('a',s) in triples[s]
                if i is not the last then:
                    REF[N_source, N_target] ← s
                else
                    add ('a',t) in triples[s]
                end if
            else
                add ('a',s) in triples[REF[source, target]]
                if i is not the last then
                    REF[N_source, N_target] ← REF[source, target]
                end if
                add ('a',t) in triples[REF[source, target]]
            end if
        else if 'classFatherOf' ∉ LABEL[source, target] AND 'subClassOf' ∉
LABEL[source, target] AND 'Inverse' ∉ LABEL[source, target] then
            if REF[source, target] then
                add (LABEL[source,target],t) in triples[REF[source, target]]
            else
                add ('a',s) and (LABEL[source, target], t) in triples[s]
            end if
            REF[N_source, N_target] ← t
        end if
    end for
end function
```

Algorithm 2 Dependency tree navigation algorithm - Part 1

```
triples ← {}                                          ▷ key-value structure
neighboor ← {}      ▷ entities that are in the vicinity of classes or instances
function ADD_NEW_TRIPLES(source, target)
    relation ← the shortest path from source to target or target to source
    if relation then:
        new_triples ← RELATION_TO_QUERY_TUPLES(relation)
        merge new_query with triples
    end if
end function
function INIT_BROTHERS(source, target)
    ref_s ← DEFINE_REFERENCE(source)
    ref_t ← DEFINE_REFERENCE(target)
    adicione ('a', source) in triples[ref_s]
    adicione ('a', target) in triples[ref_t]
end function
function EXTRACTION_RELATION(dependency_tree)
    word ← dependency_tree[WORD]    ▷ word that is at the root of the tree
    if ((word is an object property AND word not in neighboor) OR (word is not a class AND word not in neighboor)) AND there are more than one class in the word subtree then
        classes ← classes that are in the word subtree
        for i in |classes| do
            source, target ← classes[i], classes[i+1]
            if word is an object property then
                domains ← get the word domains
                ranges ← get the word ranges
                if source in domains and target in ranges then
                    relation ← the shortest path from source to target passing through word
                else if target in domains and source in ranges then
                    relation ← the shortest path from target to source passing through word
                else
                    relation ← the shortest path from source to target or target to source passing through word
                end if
            else
                relation ← the shortest path from source to target or target to source
            end if
            if relation then
                new_triples ← RELATION_TO_QUERY_TUPLES(relation)
                merge new_query with triples
            end if
            for child in children of word do
                EXTRACTION_RELATION(child)
            end for
        end for

```

Algorithm 3 Dependency tree navigation algorithm - Part 2

```
    else if word is an object property AND word has no classes in its subtree AND word in neighboor then
        source ← get URI of the class of the neighboor[word]
        ref ← DEFINE_REFERENCE(source)
        ancestors ← get the ancestors of source from the graph representing ontology
        domain, range ← get the domain and range of word
        if word hasn't been added in triples yet then
            ref_range ← DEFINE_REFERENCE(range)
            if domain == source OR domain in ancestors then
                add (URI's word, ref_range) in triples[ref]
            else if range == source OR range in ancestors then
                add (URI's word, ref) in triples[ref_range]
            end if
        end if
    else if word is an object property AND word has classes in subtree AND word has classes in ancestors nodes then
        source ← get the URI of the first word ancestor nodes
        for child in children of word do
            if child is a class OR child is an instance of the ontology then
                target ← get child URI
                if source and target are brothers in ontology then
                    INIT_BROTHERS(source, target)
                else
                    ADD_NEW_TRIPLES(source,target)
                end if
                EXTRACT_RELATION(child)
            else
                EXTRACT_RELATION(child)
            end if
        end for
        if word has not subtree then
            ancestor ← get first ancestor of word
            for brother in children if ancetor do
                if brother != word then
                    if brother is a class then
                        neighboor[brother] ← word
                    end if
                    target ← get URI of brother
                    ADD_NEW_TRIPLES(source,target)
                end if
            end for
        end if
```

Algorithm 4 Dependency tree navigation algorithm - Part 3

```
    else if word is a class OR word in neighboor OR word is a instance of ontology then
        if word is an instance of ontology OR word is a classe then
            source ← get URI of word
        else
            source ← get URI of neighboor[word]
        end if
        ancestors ← get ancestors of word
        if ancestos then
            for ancestor in ancestors do
                if ancestor is not a class AND ancestor not in neighboor then
                    neighboor[ancestor] ← word
                end if
            end for
            if word has not subtree AND word is a class then
                target ← source
                ancestor ← first ancestor of word
                if ancestor in neighboor then
                    source ← get URI of neighboor[ancestor]
                end if
                if source != target AND source and target are brothers in the ontology then
                    INIT_BROTHERS(source, target)
                    add datatype properties in variable of target and source in triples
                end if
            else
                for child in children of word do
                    if word is a class OR word is an instance then
                        mention ← word
                    else if neighboor[word] is a class then
                        mention ← word ← neighboor[word]
                    end if
                    if mention then
                        neighboor[child] ← mention
                    end if
                    if child is a class OR child is a instance then
                        target ← get URI of child
                    end if
                    if source != target AND source and target are brothers in the ontology then
                        INIT_BROTHERS(source, target)
                        add datatype properties in variable of target and source in triples
                    else
                        ADD_NEW_TRIPLES(source, target)
                    end if
                end forEXTRACTION_RELATION(child)
            end if
```

Algorithm 5 Dependency tree navigation algorithm - Part 4

```
        else
            for child in children word do
                if word is a class then
                    neighboor[child] ← word
                else if word in neighboor then
                    neighboor[child] ← neighboor[word]
                end if
                if child is a class OR child is a instance of the ontology then
                    target ← get URI of child
                    if source != target AND source and target are brothers in the ontology then
                        INIT__BROTHERS(source, target)
                        add datatype properties in variable of target and source in triples
                    else
                        relation ← the shortest path from source to target or target to source
                        if relation then
                            new__triples ← RELATION__TO__QUERY__TUPLES(relation)
                            merge new__query with triples
                        end if
                    end if
                    EXTRACTION__RELATION(child)
                elseEXTRACTION__RELATION(child)
                end if
            end for
        end if
    else
        ancestrais ← take the word ancestors
        for ancestral in ancestrais do
            if ancestral in neighboor then
                neighboor[word] ← neighboor[ancestral]
                break
            end if
        end for
        for child in children of word do
            EXTRACTION__RELATION(child)
        end for
    end if
```

B
Questions used in the evaluation of the related question generation mechanism

Table B.1: Part 1

Main question	Related questions
Quais os 5 filmes de maiores durações?	P1.1 - Quais as 5 Séries de TV de maiores durações? P1.2 - Quais os 5 Gêneros de maiores durações? P1.3 - Quais os 5 filmes de menores durações? P1.4 - Quais as 5 Séries de TV de menores durações? P1.5 - Quais os 5 Gêneros de menores durações?
Quais os 5 filmes que tiveram as maiores receitas bruta?	P2.1 - Quais as 5 Séries de TV que tiveram as maiores receitas bruta? P2.2 - Quais os 5 Gêneros que tiveram as maiores receitas bruta? P2.3 - Quais os 5 filmes que tiveram as menores receitas bruta? P2.4 - Quais as 5 Séries de TV que tiveram as menores receitas bruta? P2.5 - Quais os 5 Gêneros que tiveram as menores receitas bruta?

Table B.2: Part 2

Main question	Related question
Quais as séries de TV mais bem avaliadas no IMDB em 2018?	P3.1. - Quais os Filmes mais bem avaliadas no IMDB em 2018? P3.2 - Quais as Atrizes mais bem avaliadas no IMDB em 2018? P3.3 - Quais os Atores mais bem avaliadas no IMDB em 2018? P3.4 - Quais as séries de TV mais mal avaliadas no IMDB em 2018? P3.5 - Quais os Filmes mais mal avaliadas no IMDB em 2018? P3.6 - Quais as Atrizes mais mal avaliadas no IMDB em 2018? P3.7 - Quais os Atores mais mal avaliadas no IMDB em 2018?

C
Mappings between QALD and IMDB questions

QALD	IMDB
Qual é a receita da IBM?	Qual a receita bruta de Transporter 3?
Quais aeroportos estão localizados na Califórnia, EUA?	Quais companhias estão Brasil?
Taiko é algum tipo de instrumento musical japonês?	Mostre me séries de tv japonesas
Dê-me todos os carros que são produzidos na Alemanha.	Me mostre todos os filmes gravados na Alemanha
Quem fundou a Intel?	Quem dirigiu o Titanic?
Em que cidade fica a sede da Air China?	Onde foi gravado o filme Tropa de Elite?
Por quais países o rio Yenisei flui?	Em quais locais foi gravado o filme The Lord of Rings?
Quais políticos eram casados com um alemão?	quais os filmes dirigidos pela Woody Allen?
Mostre-me trilhas para caminhadas no Grand Canyon, onde não há perigo de inundações repentinas.	Quais prêmios sylvester stallone ganhou?
Qual é o menor jogador ativo da NBA mais curto?	Qual os 5 filmes mais mal avaliados?
Quem se tornou presidente após a morte de JFK?	Quais atores fizeram parte do elenco de The Lord of Rings?
Quantas calorias tem uma baguete?	Qual o orçamento planejado de Game of Thrones?
Quem criou a Wikipedia em inglês?	Quem foi o produtor do filme 300?
Quantos netos Jacques Cousteau tinha?	quais os filmes dirigidos pela angelina jolie?
Qual software foi publicado pela Mean Hamster Software?	Quais os filmes lançados em 2010?
Quantos imperadores a China tinha?	Mostre os diretores de filmes da China

QALD	IMDB
Qual é o rio mais longo da China?	Qual o filme mais longo?
Qual é a atmosfera da Lua composta?	Quem faz parte do elenco de Moonlight?
Quais rios correm para o Mar do Norte?	Qual o elenco de Sea North?
Quais os países da União Europeia que adotaram o euro?	Quem produziu a maioria dos filmes de ação?
Quem é o prefeito de Berlim?	Quem foi o editor de The Matrix Revolution?
Qual é o nome de nascimento de Angela Merkel?	Qual o nome da companhia que produziu o filme Slumdog Millionaire?
Me dê todas as chanceleres alemãs.	Quais atrizes fizeram parte do elenco de The Lord of Rings?
Quais são os apelidos de São Francisco?	Qual o gênero de Dexter?
Dá-me todas as ilhas frísias que pertencem à Holanda.	Liste os 5 filmes de horror mais bem avaliados?
Qual poeta escreveu mais livros?	Quais os 3 roteiristas que escreveram a maioria dos filmes?
Quem eram os pais da rainha Victoria?	Quem são os atores mais velhos?

QALD	IMDB
Quais países têm lugares com mais de duas cavernas?	Quais os 3 roteiristas que escreveram mais filmes?
Dê-me os sites de empresas com mais de 500000 funcionários.	Quais companhias gravaram mais de 700 filmes?
Quais países europeus possuem uma monarquia constitucional?	Quais os filmes europeus mais bem avaliados?
Onde Abraham Lincoln morreu?	Onde foi gravado o filme The life of Abraham Lincoln?
Qual programa de passageiro frequente tem mais companhias aéreas?	Quais os 5 atores que atuaram mais vezes em filmes?
Qual é o vulcão mais alto da África?	quais filmes feitos na América Latina mais receberam premiação?
Quando o cenário de pior caso vai ser nos cinemas na Holanda?	Quais filmes foram gravados nos Paíse Baixos?
Me dê todos os filmes com Tom Cruise.	Liste todos os filmes que Lincoln Plumer atuou
Quantos lugares tem o estádio do FC Porto?	Quais os 10 atores que mais aturam em filmes produzidos pela Athos Films?
Quais sondas espaciais foram lançadas em órbita ao redor do sol?	Quais produtoras atuam fora dos EUA?
Quem é o governador do Texas?	Quem é o diretor do filme Casablanca?
Qual é o lugar mais alto de Karakoram?	Qual o filme de drama com a melhor avaliação em 2011
Quem se chamava Scarface?	Quem foi o Produtor de Scarface?
Qual é a profissão de Frank Herbert?	Johnny Depp participou de A Star Is Born?
Quais artistas nasceram na mesma data que Rachel Stevens?	Quais os 5 atores que atuaram em filmes de maiores durações?
Em que país nasceu Bill Gates?	De qual região é a companhia warner bros?
Me dê todas as naves espaciais que voaram para Marte.	Me mostre todas as produções gravadas na França
Quantas medalhas de ouro Michael Phelps ganhou nas Olimpíadas de 2008?	Quais as 5 atrizes mais venceram Oscars nos últimos 30 anos?
Quando a empresa De Beers foi fundada?	Quando foi o início da produção de the fighter?

QALD	IMDB
Existem castelos nos Estados Unidos?	Quais os locais houveram mais gravações de filmes?
Dê-me uma lista de todos os pássaros criticamente ameaçados.	Mostre todas as atrizes que atuaram em Schindler List?
Butch Otter é o governador de qual estado dos EUA?	Vivien Leigh atuou no filme Gone with the Wind?
Quando foi fundado o Jack Wolfskin?	Quando foi o lançamento de Pirates of the Caribbean?
Me dê todos os bandidos da época da proibição.	Me dê todos os atores de Gangster escrito por Dev Anand
Quais presidentes americanos estavam no cargo durante a Guerra do Vietnã?	Qual companhia gravou a série de TV How i met your mother?
ual é a montanha mais alta ı Alemanha?	Qual o filme mais bem avaliado da Alemanha?
ual livro tem mais páginas?	Qual filme possui mais oscars?
ual é o comprimento de onda › Indigo?	Quantos atores trabalharam em The Sound of Music?
esposa do presidente Obama se ıama Michelle?	O filme Star Wars: Clone wars é uma produção de Lucasfilm ?
uando a princesa Diana morreu?	Quando foi lançado o filme Fight Club?

QALD	IMDB
Quem é o jogador mais jovem da Premier League?	Quem são os atores do elenco de Premier League poker?
Quais surfistas profissionais nasceram nas Filipinas?	Quem roteirizou o filme Filipinas?
Dê-me os sites oficiais dos atores do programa de televisão Charmed.	Me dê o nome de batismo dos atores do show de televisão Friends
Quem matou César?	Quem estrelou Al Capone?
Qual é o fuso horário do Salt Lake City?	Onde foi gravado Lake City?
Quantas empresas foram fundadas pelo fundador do Facebook?	quantos filmes foram produzidos pela angelina jolie?
Quando o criador de Drácula morreu?	Quando foi o início da produção do filmes dirigidos pela angelina jolie?
Onde começa Piccadilly?	Onde foi gravado Piccadilly?
Com quem é a filha de Robert Kennedy casada?	Quantos filmes de amor são lançados por ano?
Quais países têm mais de dez vulcões?	Liste 5 roteirista de filme de 2016
Quem foi o pai da rainha Elizabeth II?	Quem é são as atrizes de Due notti con Cleopatra?
Quando o Paraguai proclamou sua independência?	Quando a série de TV Family Guy foi indicado ao Emmy?
Quais filmes Kurosawa dirigiu?	Quais filmes foram dirigidos por Steven Spielberg?
Quantos prêmios tem Bertrand Russell?	Quantos prêmios o filme Amadeus ganhou?
Quais cidades alemãs têm mais de 250000 habitantes?	Em quais regiões foram gravados mais de 700 filmes?
Qual é o tamanho do diâmetro da Terra?	Qual a duração de Earth?
Quantos rios e lagos existem na Carolina do Sul?	Quantos atores atuaram em filmes que foram produzidos por companhia americanas?
Quem se chamava Rodzilla?	Quem editou Godzilla?
Pamela Anderson é vegana?	Em quais filmes Pamela Anderson atuou?
Quais instrumentos o Cat Stevens toca?	Quais as produções de Hans Zimmer?
Com quem é a filha de Bill Clinton?	Quais atores foram dirigidos por Woody Allen que tambem são diretores?

QALD	IMDB
Quem era a esposa do Presidente Lincoln?	Quais atores foram dirigidos por Woody Allen?
Dá-me todos os animais que estão extintos.	Mostre me os atores que fizeram filmes depois desde 2010
Quando foi Carlo Giuliani baleado?	Quando o ator Billy Wilder ganhou um Oscar?
Quantos anos o Ford Model T foi fabricado?	Há quantos anos foi lançado o filme Eternal Sunshine of the Spotless Mind?
Sean Parnell era o governador de qual estado dos EUA?	Sandra Bullock ganhou o oscar de melhor atriz em 2010?
Quem é o autor do WikiLeaks?	Quem é o figurinista do filme Wizards of the Lost Kingdom II?
Quem faz a voz de Bart Simpson?	Quem é o produtor de The Simpsons?
Quais idiomas são falados no Paquistão?	Em que linguagens City of God foi lançado?
Pelo que Elon Musk é famoso?	Jackie Chan atuou em The Tuxedo?

QALD	IMDB
Quais são os nomes das Teenage Mutant Ninja Turtles?	Qual é o gênero de Teenage Mutant Ninja Turtles?
Qual é o verdadeiro nome do Batman?	Qual foi orçamento de Batman?
Quando Michael Jackson morreu?	Quando a série Breaking Bad foi lançada?
Quantas luas Marte tem?	Quantos atores participaram do filme The Silence of the Lambs?
Me dê todas as festas holandesas.	quais filmes foram feitos na Polônia?
Me dê as capitais de todos os países da África.	Liste todos os filmes de Drama feitos por HBO
Quais surfistas profissionais nasceram na Austrália?	Me dê todas as Séries de Tv da Austrália
Quantos cientistas se formaram em uma universidade da Ivy League?	Quantos roteiristas escreveram o filme 24 Hours?
Quais idiomas são falados na Estônia?	Quais filmes foram gravados na Estônia?
Qual é o código de área de Berlim?	Qual o filme de comédia mais bem avaliado?
Com quem Lance Bass se casou?	Em quais filmes Chuck Norris atuou?
Quem é o dono de Aldi?	Quem dirigiu Aldis?
Qual é o maior estado dos Estados Unidos?	Qual o maior filme produzido nos Estados Unidos?
Qual estado dos EUA tem a maior densidade populacional?	quais companhias de filmes da alemanha que ganharam premiações?
Em que cidade nasceu o presidente do Montenegro?	De qual região é a companhia Columbia Pictures?
Me dê todos os lados B dos Ramones.	Me dê todos os filmes da Marvel Studios.
Quais empresas de fabricação de cerveja estão localizadas na Renânia do Norte-Vestfália?	Quais companhias estão localizadas na Oceania?

QALD	IMDB
Quais livros foram escritos por Danielle Steel?	Quais filmes foram escritos por Diablo Cody?
Qual é o apelido de Bagdá?	Qual o nome do filme mais bem avaliado da companhia Paramount?
Onde está localizado o Fort Knox?	Onde foi gravado o filme Fort Knox?
Me dê atores ingleses estrelando Lovesick.	Me mostre os atores que estrelaram Lovesick
Dê-me todos os escritores que ganharam o Prêmio Nobel de literatura.	Quais companhias brazileiras gravaram filme de amor e ação
Quais países estão conectados pelo Reno?	Quais filmes os atores Harrison Ford e Helen Greene fizeram juntos?
Quem escreveu Harry Potter?	Quais as 5 companhias mais ganhou o oscars?
Qual a profundidade do Lago Chiemsee?	Quão longo é o filme Ben Hur?
Me dê todos os elementos químicos.	Liste todos os filmes de comédia de janeiro de 2004
Quem estava na missão Apollo 11?	Quem são os atores de atuaram em Apollo 13?
Qual é o nome da universidade onde a esposa de Obama estudou?	Quais os nomes dos filmes em que Dwayne Johnson trabalhou?
Qual cientista da computação ganhou um Oscar?	O filme The Scientist ganhou Oscar?
Qual subsidiária da TUI Travel atende Glasgow e Dublin?	Quais produtoras brasileiras já gravaram filmes de amor e ação?

QALD	IMDB
Quais atores atuam em The Big Bang Theory?	Quem fez parte do elenco da série The big bang theory?
Dê-me todas as bibliotecas estabelecidas antes de 1400.	Liste-me todas séries lançadas antes do ano 2000
Quando foi o Boston Tea Party?	Em que região foi produzido V for Vendetta?
Quando a Finlândia se juntou à UE?	Quando foi lançado Zombieland?
Quem é o romancista do trabalho, uma canção de gelo e fogo?	Quem é o roteirista de The Devil Wears Prada?
Quando Mohamed morreu?	Quando Mohammed foi lançado?
Quem pintou A tempestade no mar da Galiléia?	Quem foi o protagonista no filme 2001:A Space Odyssey?
Quais pontes são do mesmo tipo que a ponte de Manhattan?	Quantos filmes foram nomeados com o mesmo título
Quantos filmes de James Bond existem?	Quantos filmes do 007 existem?
Como eram os nomes dos três navios por Colombo?	Em quais regiões Jackie Chan atuou?
Por quais países o rio Yenisei flui?	Quais os 10 filmes mais bem avaliados gravados pela Companhia Columbia Films?
Quais animais estão criticamente ameaçados?	Quais atrizes já atuaram em filmes e em series
Com quem Tom Hanks foi casado?	Quem foram os roteiristas de filmes que o Tom Hanks participou?
Quem é o marido de Amanda Palmer?	Quem é o roteirista de Between the Flags?
Dê-me a página inicial da Forbes.	Mostre as companhias de cinema com mais de 5 filmes feitos no Brasil
Em que estúdio os Beatles gravaram seu primeiro álbum?	Em qual companhia foi produzido o primeiro filme do diretor Martin Scorsese?
Quem são os pais da esposa de Juan Carlos I?	quem são os figurinistas de filmes escritos por Rasheem Johnson?
Qual museu exibe O Grito de Munch?	Quais são as séries tem mais de 100 pessoas no elenco?

QALD	IMDB
Qual estado dos EUA tem a abreviatura MN?	Qual o nome do filme como o maior elenco de todos os tempos?
Quais classes o Millepede pertence?	Qual produtora fez o filme Ferris Bueller Day Off?
Qual estado dos EUA foi admitido mais recentemente?	Quais os 10 filmes dos Estados Unidos mais recentes?
Quem foi influenciado por Sócrates?	Quem dirigiu o filme Socrates?
Me dê todos os filmes argentinos.	Me mostre todos os filmes argentinos
Qual é o ano de fundação da cervejaria que produz a Pilsner Urquell?	Qual foi o ano de início da produção de Heinz: Diner?
Como Michael Jackson morreu?	Quem dirigiu The Michael Jackson show?
Qual foi o nome da famosa batalha em 1836 em San Antonio?	Qual o nome do filme de maior bilheteria estreado em 1997?
Quais filhas de condes britânicos morreram no mesmo local em que nasceram?	Quais atrizes ganharam o BAFTA nos ultimos 10 anos?
Me dê todos os taikonautas.	Mostre me todos diretores dos filmes gravados na china
Quais monarcas do Reino Unido eram casados com um alemão?	Quais filmes foram gravados na Alemanha por companhias dos estados unidos?
Me dê todos os presidentes americanos dos últimos 20 anos.	Me dê todos os filmes americanos dos últimos 20 anos
Quais empresas trabalham na indústria aeroespacial e na medicina?	Quais companhias gravaram Medicinal?
Mostre-me todos os filmes checos.	mostre me todos filmes da Rússia.
Me dê todos os skatistas profissionais da Suécia.	quais filmes Rasheem Johnson escreveu?

D
Participants' Comments (in Portuguese)

Table D.1: Comments of Participants in Group 1

Comments
Não acho que filmes tenham muita relação com séries e os conjuntos P1 e P2 me pareceram muito similares.
Acho que faz mais sentido colocar a pergunta 'oposta' (mesmo assunto, trocando maior por menor e vice-versa) como primeira sugestão. Acho que filme/séries de TV estão fortemente relacionados, mas gênero não. Atriz/Ator também, seguidos por filmes/séries e, por fim, gênero. Gênero parece o mais 'solto' do grupo
Para o conjunto P2 a ordem das perguntas pareciam estar bagunçadas. Na pergunta "Quais os 5 filmes que tiveram as maiores receitas bruta?" o sistema sugeriu a pergunta relacionada "Quais os 5 filmes que tiveram as menores receitas bruta?" por último, sendo que em minha concepção deveria ser a primeira. Já para a pergunta principal: "Quais os 5 filmes de maiores durações?" ele sugeriu a pergunta relacionada "Quais os 5 filmes de menores durações" primeiro. Ou seja, o sistema que sugere o conjunto P2 parece ordenar as perguntas aleatoriamente. Para o conjunto P1 claramente pode-se perceber que as perguntas seguem uma ordem, mesmo essa ordem não sendo a melhor ordenação possível. Dessa forma, de modo geral achei o conjunto P1 melhor. Entretanto, se para a pergunta "Quais os 5 filmes de maiores durações?" a ordem de perguntas relacionadas para fosse a seguinte: 1. Quais os 5 filmes de menores durações? 2. Quais as 5 Séries de TV de maiores durações?3. Quais os 5 Gêneros de maiores durações?4. Quais as 5 Séries de TV de menores durações?5. Quais os 5 Gêneros de menores durações?A ordenação estaria perfeita! Pode-se estender esse modo de ordenação facilmente para as demais perguntas. Espero ter contribuído. Abraços :)
Achei um pouco confuso a escala utilizada de fortemente relacionada. Por exemplo, uma coisa pode estar fortemente relacionada, mas de forma negativa.
O Gênero do filme é muito dependente da pessoa que responde. Isso pode tender muito a resposta para um grupo de pessoas que você escolheu para responder e tem o mesmo convívio e participa do mesmo círculo social.

Table D.2: Comments of Participants in Group 2

Comments
Não entendi o que seria preferência.
Não entendi muito bem a finalidade do estudo. Para mim, series e filmes não são relacionados. Fiquei insegura em varias das minhas respostas.
Eu acho que me confundi sobre o que é "estar relacionado", tinham vezes que eu me questionava sobre o que era ou não um relacionamento forte. Por exemplo na primeira pergunta "Quais os 5 filmes de maiores durações?" eu dei 5/7 para a opção "Quais os 5 filmes de menores durações?", eu acho que essa opção pode estar fortemente relacionada (7/7) dependendo da definição de relacionamento forte, mudou bem pouco, só a ordenação (top 5 para bottom 5). Mas como não sei ao certo o que define um "relacionamento", preferi ser conservador e assumir que relações máximas são como paráfrases, tem que ter a mesma informação escrita de forma diferente e qualquer mudança na informação é penalizada. Acabei percebendo que no final das contas criei uma regra, mudanças de ordem (como no "maior para menor") eu penalizei mais do que mudanças de contexto (como de "filmes para séries"). Não sei se era essa definição de relacionamento que estava esperando. Acho que na próxima vale a pena dar um background do que seria um relacionamento entre perguntas e dar uns exemplos de variações na escala de forte a fraco. Não sei se a sua ideia era de fato deixar a galera no escuro mesmo e ver como era o entendimento geral sobre o que seria um relacionamento forte/fraco, se for isso já te adiantei qual foi o modelo mental que usei. Agora só torcer pro meu entendimento não ser um outlier :)
pra mim a vocês poderiam usarpoderia usar ou (1) o objeto ou (2)a direção (maior/menor) como critério de ordenamento. o primeiro conjunto é nenhum desses e o segundo conjunto ordena usando (maior/menor) como critério primário preservando o objeto como secundário.
Acredito que quando ha menos alterações, na ordem, de uma frase para outra fica mais fácil de identificar/perceber as mudanças/semelhanças.
Talvez fosse melhor colocar para esse tipo de votação ser feito individual por pergunta. É difícil avaliar da forma que está.
Acho o conceito de gênero muito vasto pra ser considerado próximo de filme