Using a Domain Ontology to Bridge the Gap between User Intention and Expression in Natural Language Queries

Alysson Gomes de Sousa[1], Dalai Dos Santos Ribeiro[1], Rômulo César Costa de Sousa[1],
Ariane Moraes Bueno Rodrigues[1], Pedro Henrique Thompson Furtado[2],
Simone Diniz Junqueira Barbosa[1] and Hélio Lopes[1]
[1]*Department of Informatics, PUC-Rio, Brazil*
[2]*CENPES - PETROBRAS, Brazil*

Keywords: Ontology, Knowledge Base, Data-driven Search, Progressive Disclosure, Question Answering.

Abstract: Many systems try to convert a request in natural language into a structured query, but formulating a good query can be cognitively challenging for users. We propose an ontology-based approach to answer questions in natural language about facts stored in a knowledge base, and answer them through data visualizations. To bridge the gap between the user intention and the expression of their query in natural language, our approach enriches the set of answers by generating related questions, allowing the discovery of new information. We apply our approach to the Movies and TV Series domain and with queries and answers in Portuguese. To validate our natural language search engine, we have built a dataset of questions in Portuguese to measure precision, recall, and f-score. To evaluate the method to enrich the answers we conducted a questionnaire-based study to measure the users' preferences about the recommended questions. Finally, we conducted an experimental user study to evaluate the delivery mechanism of our proposal.

1 INTRODUCTION

Search has become ubiquitously associated with the Web, to the point of becoming a default tool in any modern browser and one of the most popular activities online, already in 2008 (Fallows, 2008).

A major challenge for search systems 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 is specifically more difficult because there's not a set of predetermined answers, as in the classification tasks (tokenization, pos-tagging, named entity recognition). This task also presents the following challenge: how can we capture the user's intention expressed in a natural language question/request and translate it into a computationally processable query? And in this case, in the Portuguese language. Many systems try to achieve this by allowing the user to navigate through search results and refine the search query. However, formulating a good query can be cognitively challenging for users (Belkin et al., 1982), so queries are often approximations of a user's underlying need (Thompson, 2002). Although most of the search systems are effective when the user has a clear vision of their interests, those systems may not be very suitable when the user is performing an exploratory search or cannot properly formulate their information need.

Instead of requiring users to manually adjust the queries to amplify their search results, our hypothesis is that a search system that continually offers answers to related queries based on navigation through an underlying domain ontology would improve the user experience. We developed a system to explore the Movies and TV Series domain, using the IMDb Movie Ontology developed by (Calvanese et al., 2017), an ontology to describe the movie domain semantically. Their ontology uses the International Movie Database data as its data source. In this paper, we focus on searches whose results can be represented as data visualizations.

2 RELATED WORK

Exploratory Search. Information seeking is well supported by search engines when the user has well-defined information needs. However, when the

de Sousa, A., Ribeiro, D., Costa de Sousa, R., Rodrigues, A., Furtado, P., Barbosa, S. and Lopes, H.
Using a Domain Ontology to Bridge the Gap between User Intention and Expression in Natural Language Queries.
DOI: 10.5220/0009412307510758
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 751-758
ISBN: 978-989-758-423-7

user lacks the knowledge or contextual awareness to formulate queries or navigate complex information spaces, the search system should support a complex information seeking process, where the user can browse and explore the results to fulfill their needs (Wilson et al., 2010).

Exploratory Search research studies information-seeking models that blend querying and browsing with a focus on learning and investigating, instead of information lookup (Marchionini, 2006). There are three typical exploratory search situations: (i) The user has partial or no knowledge of the search target; (ii) The search moves from certainty to uncertainty as the user is exposed to new information; and (iii) The user is actively seeking useful information and determining its structure (White et al., 2005).

O'Day and Jeffries describe an incremental search behavior as a process of exploration through a series of related searches on a specific topic (O'Day and Jeffries, 1993). They identify three distinct search modes: (i) monitoring a well-known topic over time; (ii) following a plan of information gathering; and (iii) exploring a topic in an undirected fashion. This shows that even exploratory information seeking has structure and continuity, which could be supported by the search system.

Semantic Web and Natural Language Processing. The volume of digitally produced data keeps growing greatly, but mostly to be read and interpreted by people. Their lack of structure and standardization makes automatic processing highly expensive or ineffective. In 1994, the Semantic Web emerged as an extension to the traditional Web that aims to make Web content processable by machines, mainly through two technologies: domain ontologies and resource description framework (RDF). Domain ontologies are a flexible model for organizing the information and rules needed to reason about data (Berners-Lee et al., 2001). In the context of Computer Science, an ontology is a formal description of knowledge of a particular domain (Gruber, 1993). RDF is a model that provides the foundation for metadata processing, making web resources understandable to machines (Lassila et al., 1998).

Another area that aims to structure unstructured data Natural Language Processing (NLP). Some NLP tasks have become highly relevant to other areas of knowledge, including the Semantic Web. Among these tasks, we can highlight the Dependency Analysis. Dependency analysis seeks to capture the syntactic structure of the text represented through dependency relations organized in a structure called dependency tree. One of the main advantages of using this dependency framework is that the relationships extracted in the analysis provide an approximation to semantic relationships. Therefore, these dependency structures are useful for extracting structured semantic relations from unstructured texts.

Query Interpretation. Several approaches for answering natural-language questions use NLP techniques and lexical features that relate words to their synonyms, together with a reference ontology. Some works integrate dependency trees into other methods and resources (Yang et al., 2015; Paredes-Valverde et al., 2015; Li and Xu, 2016). The first uses vector representations to capture lexical and semantic characteristics, and 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. The third proposes an approach that uses the identified entities and navigates the dependency tree guided by these entities.

Another challenging aspect of question answering 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. To resolve this problem, many works ask the user for the correct interpretation whenever there is ambiguity (*e.g.*, (Melo et al., 2016; Damljanovic et al., 2012; Yang et al., 2015)).

3 DEFINING AND ENRICHING THE DOMAIN ONTOLOGY

Domain ontologies have been popularized as a flexible knowledge representation model for organizing the information and rules needed to reason about data (Berners-Lee et al., 2001). In an ontology, formally declared real-world objects and relationships between them form the *universe of discourse*, which reflects the domain vocabulary and thus the knowledge the system will have about that domain (Gruber, 1993). The definition of domain ontology is an important part of the process. The ontology should contain the main classes and individuals of the domain, and well-defined relationships with respective domains and ranges. In this paper, we used a movie ontology that describes the cinematography domain developed at the Zurich University CS department[1].

[1]https://github.com/ontop/ontop/wiki/Example_Movie Ontology

When searching for information, a user may often not know how to express what they want. They start from an initial set of terms and, from the search results, they will refine the search to improve results, as pointed out by (Marchionini, 1997). Many models provide information related to the initial results, to reduce the user's cognitive effort to formulate new terms that refine or broaden the results of the search (Sun et al., 2010; Setlur et al., 2016; Gao et al., 2015).

Our approach enriches the answer set using relationships identified in the user question and ranked by the strengths of those relationships. The ontology is enriched with annotations that define relationships that the user finds interesting, given that certain entities were cited in the initial question. The terms defined in the annotation ontology were the following:

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

Other ontologies may reuse these annotation terms similarly to other annotation properties defined in the OWL vocabulary. An exception is the *relationshipStrength* annotation, which is defined as a *Datatype Property*, because it is a list composed of two related entities with a numeric value representing the force of relationship. We have also created a property called *isBasicLevelCategory*, 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.

4 OUR APPROACH

Our approach comprises question interpretation, query expansion, and result delivery, described next.

4.1 Question Interpretation

We needed a mechanism capable of answering natural language questions in Portuguese based on an ontology and a knowledge base. It must be able to capture the intention or desire expressed in a user question or request and convert it into a SPARQL query.

The first step in the interpretation process is **entity detection**, which consists of identifying the classes, properties, and individuals expressed in the reference ontology and knowledge base mentioned in the question or request for information. We assume that all classes, properties, and individuals are annotated with the *label* property, defined in the standard RDF vocabulary, preferably set to Portuguese (*i.e.*, adding the suffix @*pt-br* after the *label* content). It is necessary to deal with variations in terms (due to verbal inflections, gender, and number). 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*.

To improve entity detection, we use a lexical feature called Onto.PT, created by (Gonçalo Oliveira and Gomes, 2014), a synonym ontology developed for Portuguese. It adopts the concept of synsets, which are sets of *synonyms*. 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.

It is common for the same terms to designate different entities, so a **disambiguation** step is required. This step checks which terms or sets of terms match more than one entity, including classes or properties of the ontology 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, for users to determine the correct option.

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. Indexing an ontology means to create a global graph that unites the hierarchy, individuals, and relationships expressed in the properties. Structuring the ontology this way was useful both in practical terms, supporting for the search for relationships, and in theoretical terms, since the concepts of graph theory (such as shortest path and neighborhood) could also be used.

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: "Quais atores participaram de filmes que ganharam o Oscar?" (Which actors participated in films that won the Oscar?). The terms *atores* and *filmes* coincide with reference ontology classes, and the term *oscar* corresponds to a named individual.

After generating the dependency tree from the question, we extract the relationships: we traverse the tree, 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 classes are related to the previous node. 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 1.

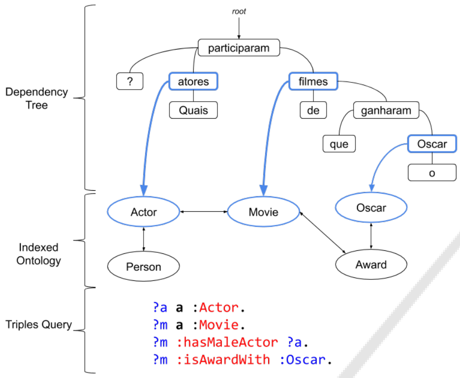

Figure 1: Relationship Extraction from Dependency Tree.

Finally, we evaluate the other terms that match the keywords defined in our method, processing them as parameters whose values are in the neighborhood of the term or its subtree.

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

4.2 Query Expansion

In order to enhance the answers generated by the interpreter and to reduce the user's cognitive effort to formulate other related questions which may interest them, our group developed a mechanism that recommends answers to questions related to the initial user question. This mechanism applies operations to the ontology, taking into consideration the entities that were detected in the initial question.

Let us take an example: the information needed by the user is the movie *genre* that generated the highest box office in 2018, but when formulating their query they typed: "Which *movies* had the highest gross revenue in 2018?". The interface sends, through an API, the natural language query written by the user. The API looks for the literal answer or answers to the question and ranks the results. It then exhibits the *n* highest ranked literal results for the query on the topmost area of the interface, in a slightly shaded area (Figure 3). Below that area, it progressively displays results for related questions, which are gradually received from the API. Those results are the outcomes of a search mechanism that, given a domain ontology (*e.g.*, IMDb), navigates through the ontology looking for useful relationships between the elements presented in the search query to expand the given question into related ones.

JARVIS may offer, for example, results for questions such as "Which *studios* had the highest gross revenue in 2018?" (through a movie–produced by–studio relationship), "Which *movies* had the highest gross revenue in 2018 per country?" (through a movie–produced in–country relationship), and "Which *movie genre* had the highest gross revenue in 2018?" (through a movie–classified as–genre relationship). These related questions may offer the information needed by the user, as well as different perspectives on the data related to the query, without any manual interaction by the user.

Once we have the ontology appropriately annotated, our strategy for generating related questions involves using the entities identified in the interpretation mechanism and rephrasing the initial question by replacing the entities mentioned with related entities (defined at the time of the annotation).

Figure 2 schematically shows a clipping of the previous ontology with their respective relationships of interest and an initial question with some identified ontology entities.

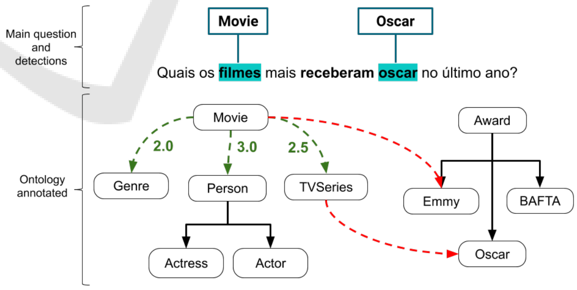

Figure 2: Sample Question and Its Relationships of Interest.

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 2, we can generate the following related questions:

- What actors received the most Oscars last year?
- What actresses received the most Oscars last year?
- Which films received the most BAFTAs in the last

year?

As the set of generated questions can become too large, because of the number of entities cited in the initial question (Duarte. et al., 2016), the ranking policy may influence the exploratory search and, thus, the knowledge discovery. Therefore, we define an ordering criterion that combines the strength of the relationship with the number of different entities in each related question, defined by Equation 1.

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

where C is set of relationships of interest and w is set of weights associated to relationships. Thus, ranked questions go from questions with few variations and high relationship strength to questions with many variations but low relationship strength.

4.3 Delivery Mechanism

Many systems, such as Datatone (Gao et al., 2015), allow the user to navigate through related questions by direct manipulation of the query or through manual interactions with its ambiguity widgets. We hypothesize that, instead of requiring users to manually adjust the queries to amplify their search results, user interfaces for searching data visualizations may continuously offer answers to related queries.

Figure 3: JARVIS Search User Interface (Model J3).

Our proposed search user interface (Figure 3), named *JARVIS - Journey towards Augmenting the Results of Visualization Search*, is based on the progressive disclosure model used by Google Images, where the interface continually appends search results to the search results page. Rather than requiring users to refine their queries, JARVIS automatically amplifies the set of results with answers to related queries.

Suppose the user wants to know the movie *genre* that generated the highest box office in 2018, but they formulate their query as: “Which *movies* had the highest gross revenue in 2018?” JARVIS sends, through an API, the natural language query written by the user. The API looks for the literal answer(s) to the question, ranks the results, and shows the n best literal results for the query on the topmost area of the interface, in a slightly shaded area. Below that area, it progressively displays results from the related questions, which are gradually received from the API.

5 EVALUATION

5.1 Query Interpretation

To evaluate our approach, we created a knowledge base from the the aforementioned IMDb ontology and the IMDb database.[2] We built a question dataset based on the Question Answering over Linked Data (QALD[3]) competition question dataset, adapting the structure and questions to the IMDb context. We first took the main question types in the QALD dataset, such as questions that have the terms *what*, *who*, *when*, *where*, then the main operations applied to queries, such as count, sorting, grouping, and temporal filtering. Finally, we associated the classes and individuals mentioned in the QALD questions with the classes and individuals in the IMDb context.

Our dataset[4] comprises 150 questions in Portuguese and English, with the SPARQL query of each question. The dataset is distributed of the following way: 94 questions of the type *what*, 25 of the type *who*, 11 of the type *count*, 9 of the type *when*, 6 of the type *yes/no*, and 5 of the type *where*. We applied our approach to each question and calculated the usual metrics: *Precision* ($mean = 0.58, var = 0.239$), *Recall* ($mean = 0.63, var = 0.231$), and *F1-score* ($mean = 0.57, var = 0.231$). On average, *Recall* $>$ *Precision*, indicating that most of the relevant answers were selected. However, part of what was selected is not relevant, evidenced by low Precision values.

However, our approach tries to respond to specific questions and not only factual questions, as other works focused on Portuguese do (Penousal and

[2] https://www.imdb.com/interfaces/
[3] https://github.com/ag-sc/QALD
[4] https://github.com/alyssongomes/dataset-questions-imdb

Machado, 2017; Teixeira, 2008; Cortes et al., 2012). This difference is important because the queries generated to answer the factual questions are generally built from a composition of keywords, sometimes amplified through synonyms, where the set of answers are results of these queries. Therefore, too many answers are part of the approximated set of results.

Specific questions require more structured queries, and the fact of the vast majority of research in NLP is conducted on the English language lead to NLP tools more accurate as for Portuguese (Otter et al., 2018), which makes the task more difficult. Table 1 shows the mean and variance of the F-score results broken down by each question type in the dataset. Here we can see that the best results came from questions where a location (*where*) or a temporal (*when*) attribute was 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 1: 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

Questions of type *what* concentrated most of the errors. This may have happened due to the wide variety of questions of that kind. This strongly impacts the dependency tree generation, as its structure can vary considerably depending on the way the question is formulated. 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*. Our approach has difficulty distinguishing questions of the type *yes/no*, usually because they are very similar to questions of type *what*, but require to determine when to test only 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, and some of the data were missing from the knowledge base, which explains these findings.

5.2 Query Expansion

To evaluate the query expansion engine, we took the IMDb ontology and arbitrarily defined a set of relationships of interest and the strengths of each relationship in the IMDb ontology. For instance Actor-Movie has strength 3, whereas Genre-Movie has strength 2. The evaluation was performed from the perspective of users who have some familiarity with search engines. For this, the user evaluated related questions generated from the following set:

P1: What are the best rated TV shows on IMDb in 2018?

P2: Which 5 movies had the highest gross revenues?

P3: What are the 5 films of the longest duration?

To evaluate these questions, the participant informed how closely they considered the generated question to be related to the initial question, in a 7-point scale (1-Not related to 7-Strongly related). To evaluate the order in which questions are shown, participants received an online form with 2 question sets, P and A, where P contained the related question groups P1, P2, and P3 in the order proposed by our approach, and A contained the same groups of question listed in random order (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. At the end of each question set, the participant evaluated the set of related questions as a whole and the order in which the related questions were listed, and chose their preferred group.

Most of the 42 participants were undergraduate and graduate students in Computer Science, Chemical Engineering, or Electronics. In Figure 4, we show the results for the questions sorted according to the criteria proposed in this paper. We can notice a trend in the evaluations: as expected, the number of scores that indicate weak relationships grows as the question has lower positions in the ranking.

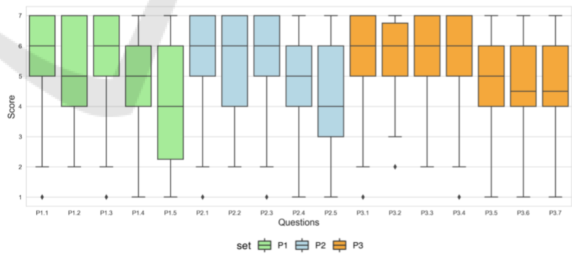

Figure 4: Compiled Distribution of Group P Assessments.

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. We found a significant difference only between P3 and A3, revealing 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. We also analyzed the impact of ordering on the scores that were given to the individual questions. A Kruskal-Wallis test showed a significant difference between the questions of each ordered group P1, P2, and P3, with the following p-values: 0.0000239, 0.0000554, and 0.0002219, respectively. A Conover-Iman post-hoc test revealed a significant difference in 4 P1 question pairs, 4 in P2, and 5 in P3.

Significant differences occurred when more than one entity was modified, so the question suffered a ranking penalty. This means that questions with more modified entities really should have lower priority. The details about of interpretation method and the results of evaluations can be queried in (Sousa, 2019).

5.3 Delivery Mechanism

JARVIS progressively discloses results for related queries. To evaluate the effectiveness and efficiency of our solution, we have devised two other search user interface (SUI) models for the same search task. The first uses the traditional search interaction method described by (Wilson, 1999) (henceforth called *Traditional* SUI (J1)), and the second is built showing the related questions as links to explore the results (henceforth called *Related-links* SUI (J2)). Then we conducted a comparison study of the three SUIs. We invited graduate students from different areas to serve as volunteer participants in the study.

In J1, the user types a search query and receives the highest ranked result for their question. The only way they can expand the search results is by manually editing or typing a new query for the system. This model represents a baseline for our work.In J2, the user is now presented not only with the highest ranked result, but also with a set of related questions on a side pane. This allows the user to navigate through related questions more quickly, but still requires manual interaction with the user interface. Model J2 is presented to the participants so we can attempt to understand whether the mere introduction of related questions is enough to reduce the users' cognitive overload and to build a more effective search interface.

To evaluate the interface models, we conducted an empirical comparative test of the three SUIs. To reduce the learning effects, we varied the order in which each SUI was presented. Fifteen people participated in the experiment to evaluate the delivery mechanism of the related queries: three female and 12 male. They were all graduate students at PUC-Rio (11 Master's and 4 PhD students). Apart from one Psychology student, all the participants were Computer Science students. All participants were familiarized with traditional search tools. Four participants had already seen a search user interface similar to JARVIS (J3) in another context, but had not used it. One participant helped develop J3 for an R&D project. The other ten participants had no knowledge of models J2 and J3.

For each SUI, the user received six search tasks, each one representing a search query. In the Related-links SUI (J2) and JARVIS (J3), participants would need to type only two queries, and then they would have quick access to the remaining related queries through the links at the right-hand panel. In the Traditional SUI (J1), however, the user would need to type in each of the six queries manually.

After interacting with each SUI, we asked the participant to fill out a questionnaire regarding the perceived ease of use and utility of the SUI based on the Technology Acceptance Model (TAM) (Davis, 1989) and their subjective workload assessment based on the NASA Task Load Index (Hart, 2006). At the end of the session, we briefly interviewed the users, asking them to choose their preferred SUI and explain why. We also collected performance data in terms of effectiveness (correctness of the result) and efficiency (time on task). In particular, we used the number of searches as a proxy for efficiency. The results showed a significant difference between J1 and the exploratory behavior of models J2 and J3. The details about of delivery mechanism and the results of evaluations can be queried in (Ribeiro, 2019).

6 CONCLUSIONS

We proposed an approach to answer questions in natural language by converting them into SPARQL queries, using a knowledge database and a domain ontology. Our model amplifies cognition for search tasks by generating and presenting related queries to expand the search space and progressively disclose the corresponding results. We evaluated the performance of the method and obtained 58% average precision, 62% average recall, and 57% average F-score.

We evaluated the proposed model of delivery (J3) in comparison with two distinct search user interface models for data visualization. Our results suggest that the proposed methodology has potential as a novel design search systems that bridge the gap between the user intentions and the queries they produce.

ACKNOWLEDGEMENTS

This study was financially supported by CAPES, CNPq, and PETROBRAS.

REFERENCES

Belkin, N., Oddy, R., and Brooks, H. (1982). Ask for information retrieval: Part i. background and theory. *J. Doc*, 38(2):61–71.

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. *Sci. Am.*, 284(5):34–43.

Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., and Xiao, G. (2017). Ontop: Answering sparql queries over relational databases. *Semantic Web*, 8(3):471–487.

Cortes, E. G., Woloszyn, V., and Barone, D. A. C. (2012). *When, Where, Who, What or Why? A Hybrid Model to Question Answering Systems*, volume 7243. Springer International Publishing.

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*.

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3):319–340.

Duarte., E. F., Nanni., L. P., Geraldi., R. T., OliveiraJr., E., Feltrim., V. D., and Pereira., R. (2016). Ordering matters: An experimental study of ranking influence on results selection behavior during exploratory search. In *Proceedings of the 18th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 427–434. INSTICC, SciTePress.

Fallows, D. (2008). Search Engine Use | Pew Research Center.

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*, UIST '15, pages 489–500, New York, NY, USA. ACM Press.

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

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

Hart, S. G. (2006). Nasa-Task Load Index: 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908.

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.

Marchionini, G. (1997). *Information Seeking in Electronic Environments*. Cambridge Series on Human-Computer Interaction. Cambridge University Press.

Marchionini, G. (2006). Exploratory search: From finding to understanding. *Commun. ACM*, 49(4):41–46.

Melo, D., Rodrigues, I. P., and Nogueira, V. B. (2016). Using a dialogue manager to improve semantic web search. *INT J SEMANT WEB INF (IJSWIS)*, 12(1):62–78.

O'Day, V. L. and Jeffries, R. (1993). Orienteering in an information landscape: How information seekers get from here to there. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 438–445, New York, NY, USA. ACM.

Otter, D. W., Medina, J. R., and Kalita, J. K. (2018). A survey of the usages of deep learning in natural language processing. *arXiv preprint arXiv:1807.10854*.

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 SYST APPL*, 42(12):5163–5176.

Penousal, F. J. and Machado, M. (2017). RAPPORT: A Fact-Based Question Answering System for Portuguese. (June).

Ribeiro, D. d. S. (2019). Exploring ontology-based information through the progressive disclosure of visual answer to related queries. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

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.

Sousa, A. G. d. (2019). An approach to answering natural language questions in portuguese from ontologies and knowledge bases. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

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.

Teixeira, J. (2008). Experiments with Query Expansion in the RAPOSA (FOX) Question Answering System. (September).

Thompson, P. (2002). Finding out about: A cognitive perspective on search engine technology and the www. *Information Retrieval*, 5(2):269–271.

White, R. W., Kules, B., and Bederson, B. (2005). Exploratory search interfaces: Categorization, clustering and beyond: Report on the xsi 2005 workshop at the human-computer interaction laboratory, university of maryland. *SIGIR Forum*, 39(2):52–56.

Wilson, M. L., Kules, B., Shneiderman, B., et al. (2010). From keyword search to exploration: Designing future search interfaces for the web. *Foundations and Trends® in Web Science*, 2(1):1–97.

Wilson, T. (1999). Models in information behaviour research. *J. Doc.*, 55(3):249–270.

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