An Information Retrieval Model using Query Expansion based on Ontologies in the Computer Science Domain

Bonnie G. Carranza Chávez*, Andrés Melgar*†
* Grupo de Reconocimiento de Patrones e Inteligencia Artificial Aplicada,
Pontificia Universidad Católica del Perú, Lima, Perú
† Sección de Ingeniería Informática, Departamento de Ingeniería,
Pontificia Universidad Católica del Perú, Lima, Perú

***Abstract*—This paper presents a model that aims to support knowledge retrieval stored in digital repositories through domain ontologies. In this model the ontology contains concepts and relationships which describe a specific part of the world. The model mechanisms aim to reduce the impact of some of the main obstacles identified in the Information Retrieval process such as user specific characteristics, natural language characteristics or retrieval systems limitations. As a result the user, by providing a query to the system, can retrieve relevant information which better meet his information need. A prototype was developed to demonstrate the feasibility of the model using queries in the computer science domain.**

Index Terms*—*information retrieval, query expansion, ontology

I. Introduction

In the past years, we have been witness to an exponential growth of digital information [1] which was triggered by the continuous development of IT. From a user's point of view, the traditional way to meet his information need would be performing an exhaustive review of contents from physical and digital documents [2]. However, this process clearly demands a considerable investment of time and effort, for that reason this alternative may not be possible for many users.

In order to lighten this process, Information Retrieval Systems (IR systems) progressively emerged [3]. These systems represented a tool for, in an automated way, retrieving useful information corresponding to the user's query, which at the same time lightened to a certain degree the difficulties of the manual exhaustive traditional search process [4]. However, nowadays these systems not necessarily present an ideal behavior. In first place, since they are not always able to effectively interpret what users want and need, it is not unusual they provide irrelevant documents. In second place, due to intrinsic characteristics of Natural Language (NL) such as: words ambiguity, context dependencies, the fact that a word may have different domain-specific meanings and the fact that a concept may be expressed by different words. As a result, it is common that documents relevant for the user are omitted, or that excessive information that does not meet user requirements is delivered.

In this paper we propose an alternative to the retrieval information problem so that retrieved documents are to a certain degree more relevant. This retrieval is treated under the Query Expansion (QE) approach for which knowledge models such as ontologies are used. In specific, we developed an ontology in the Computer Science (CS) domain in the scope of a university curricula and a prototype to test the model. After analyzing the results of tests, it was concluded the integration of components succeeded on retrieving information relevant for the user and overcoming to a degree some of the obstacles identified for the retrieval process.

II. Literature Review

A. Information Retrieval

IR can be understood as the scientific discipline in charge of the analysis, design and implementation of computer systems which deal with the representation, storage, organization and access to non-structured information, and can provide answers to user queries [5]. The retrieved information which from the user point of view meet the stated query is called "relevant". The IR discipline focus on the maximization of retrieved relevant documents while at the same time minimizing the retrieval of non-relevant documents. These objectives can be quantified through the use of precision (ratio of the number of relevant documents retrieved to the total number of documents retrieved) and recall (ratio of the relevant documents retrieved to the total number of relevant documents) metrics [6].

B. Query expansion

Usually, users tend to formulate short queries instead of carefully built ones. Such short queries lack of words that if were provided, could be very useful search terms [7]. The QE goal is to add new meaningful terms to the initial query [8]. For example, for a query stating *Pilas* which is an ambiguous plural-form Spanish word that may refer to batteries, cells, heaps and stacks, adding the word *Baterias* (batteries) to the query would be meaningful because it would help the system to identify the domain the user is trying to query about. This addition would represent a QE.

C. Ontologies in IR

A conceptualization is an abstract, simplified view of the world. An ontology is an explicit specification of a conceptualization [9]. It consists of entities, attributes, relationship,

(DOI reference number: 10.18293/SEKE2015-002)

and axioms in a human understandable and machine readable format [10]. In recent years, ontologies have been adopted in many business and scientific communities as a way to share, reuse and process domain knowledge. For example, an ontology in the animal diseases domain developed by a medical expert would represent a base of knowledge which could be used by software developers to create applications to diagnose an animal illness from the symptoms [11]. Ontologies are increasingly being used in IR research as knowledge to support semantic search [12], [13]. For an ontology based IR system, when the user inputs the QE, the system tries to insert the ontology knowledge to enhance the QE in order to increase the probability of relevancy [10]. In [14] authors discuss that it isn't optimal the using of general purpose ontologies like WordNet for specific domains because it could lead to the losing of precision. For that reason, they introduced a QE algorithm for medical IR using concepts from the MeSH ontology. A different approach was proposed in [10], where the author introduced ontologies into QE and made a deep use of semantic relations of concepts to expand query keywords and to make the retrieval results more accurate. In [15] authors used automated QE with the support of ontologies. The objective of their QE in data integration proposal is to extend the results of a given query in a semantically meaningful way. They focused in the integration of different sources, and over this unification performing the QE.

III. Proposed Model for Information Retrieval

The proposed model (see figure 1) was designed to facilitate IR using both ontologies and user information as input for the QE. It consists of 5 layers with a total of 10 components:

- **Visualization Layer**: is used to get the input from user and to show the outputs.
- **Support Layer**: contains support components in charge of coordinating all interactions and preparing the query for the expansion.
- **Retrieval Layer**:retrieves the documents based on the expanded query and the tagged documents.
- **Expansion Layer**: in charge of coordinating the overall QE process. This component has two sub-components, the `Equivalence Handler` and the `Ambiguity Handler`.
- **Data Access Layer**: retrieves information from the ontology, the documents repository and the user information DB.

A. Preprocessing Handler

This component aims to reduce the difficulties determined by the difference between how is knowledge stated in documents and how it was formulated in the query. In order to reduce some of the blurring effects of NL characteristics over the retrieval effectiveness two mechanisms are proposed. The first one, the stopwords removal mechanism (SRM), will perform a removal of those words which do not make a significant contribution of relevant concepts. The second one, the lemmatization mechanism (LM), will support the simplification of both, the query and the documents tags, in order to reduce different representations of a same concept to a single base form.

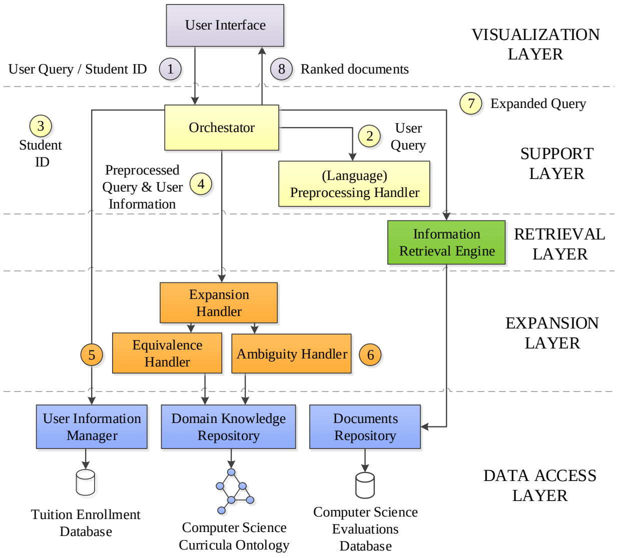

Fig. 1. Model architecture

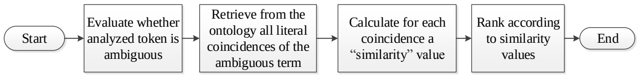
Start
Evaluate whether analyzed token is ambiguous
Retrieve from the ontology all literal coincidences of the ambiguous term
Calculate for each coincidence a "similarity" value
Rank according to similarity values
End

Fig. 2. General flow of the disambiguation mechanism

B. Expansion Handler

This component is in charge of performing the search, selection and addition of significant new terms to the query. It is supported by two sub-components which aim at dealing with two different NL features.

1) Equivalence Handler: Given that a same concept can be expressed by different words, this component aims to enrich the query by adding different but semantically equivalent words. Has as input the resulting terms from the preprocessed query. In first place, a list of nodes which does not contain the analyzed term will be retrieved.Then, a search of that term within the synonyms of the retrieved nodes will be performed, and if the term is found in the synonyms, both, synonyms and main concept, will be added to the expanded query.

2) Ambiguity Handler: This component will allow to identify the related concept of one or more words considered ambiguous within a knowledge domain. A domain ontology and the other supplementary query terms will support the identification of possible concepts to which the ambiguous term in analysis could be making reference. Figure 2 shows the general flow for the proposed mechanism, which was designed adapted from the general flow of the personal name disambiguation model [16]. The similarity computing consists of a recursive evaluation of the ontology nodes, and at each level evaluating if the supplementary query term is found in the analyzed node or in its equivalent (*i.e.*, synonym nodes). For cases when the retrieved coincidence itself contains the supplementary term, the similarity is maximum (defined as 0), otherwise, the evaluation will continue in the hierarchically superior nodes which have a relationship with the analyzed (ambiguous term-coincident) node. The similarity value is defined as $CONSTANTEREC * LEVEL$, where level is the recursiveness level where the supplementary term was found. The more distant levels will be considered as less similar, and this will represent the end of the similarity calculation. Another stop condition is set to when reaching a maximum number of recursions without having found a coincidence of the supplementary term at any analyzed level. Finally, a ranking will be established based on the calculated scores. The top ranked term (similarity nearest to 0) represent the most accurate concept which will be added to the expanded query. The purpose of the $CONSTANTEREC$ constant will be described in the User Information Manager section.

C. Domain Knowledge repository

In order to represent and store the knowledge, a customized ontology in a university curricula in the CS domain was developed with the following considerations:

- A property `NombrePreferente` (Preferred Name), designed to store the principal name of each concept. Each node has only one preferred name.
- A property `Sinonimos` (Synonyms), which relates each node to their equivalent lemmatized terms without stop-words. Each node may have one or more synonyms.
- A property `Lemma`, which relates each node to their respective base form denomination. Each node in the ontology has exactly one associated lemma.

Even though each node's `NombrePreferente` is fulfilled based on the specialized knowledge provided by an expert, the lemmatized forms to be put in the other two properties are generated using the preprocessing mechanisms previously explained. After obtaining those lemmatized terms, they are manually inserted in the ontology development phase because the preprocessing mechanisms' overall execution time, when applied in a complex structure like this ontology is considerable, so online execution is not feasible.

D. Documents Repository

For simplification purposes, for this work it has been excluded from the scope the use of online information, and instead the documents repository consist of documents tagged with semantic content inserted to a relational DB.

E. User Information Manager

It consists of artifacts related to the information about the user. For our case the user information DB is a relational DB representing a tuition enrollment management system for university students, which contains information about the courses each student is currently enrolled in. It is within the disambiguation mechanism that user information will increase in relevance, because in case two or more concepts obtain the same similarity score, the user information will be used to route the decision to one concept. In order to do so an additional flow is added to the similarity calculation which consist of verifying if the analyzed node is present in the user information and if so, decrementing the score in order to get it closer to a better similarity score. When using the user information the $CONSTANTEREC$ constant increase in relevance, because if the space enabled by the use of this constant would not exist, it may be the case that when decreasing the similarity score of a concept a new tie occurs.

IV. Prototype Implementation

In order to demonstrate the feasibility of the proposed model, we developed a prototype. The SRM was built based on the `StopAnalyzer` from Lucene library, and was configured with the default stopwords dictionary used by Lucene's `SpanishAnalyzer`. The LM was built based on the default Spanish lemmas dictionary from the Freeling language analysis tool. The disambiguation mechanism was built with support of the SPARQL query language for navigation and retrieval from the ontology. As part of the configuration for the prototype, the stop for the recursion was set to a deep of up to 5 levels, and the $CONSTANTEREC$ constant value was set to 3. The user information DB was filled with 30 tagged documents including university examinations in the CS domain and the IR Engine was developed based on Lucene without particular customizations. The developed ontology was created using Protégé in OWL/RDF.

V. Results

As previously explained, the ontology was developed considering knowledge on a CS curricula. We took a query stating *pilas y quicksort*, in an attempt to retrieve information regarding to the specific topics of `Stacks` (abstract data type) and the `Quicksort` algorithm. The word *Pilas* is an ambiguous plural-form Spanish word. The word *Quicksort* even though it does have a Spanish translation, the reference to the sorting algorithm can be found indistinctly either in English or in Spanish. When the user inputs the query (fig. 1 step 1), the SRM removes the stopword *y* (and) and the LM converts the resulting *pilas quicksort* query to *pila* (single form) *quicksort* (step 2). Next, the student ID provided by the user is sent to the `User Information Manager` in order to retrieve the courses he is currently enrolled in (step 3). Then, the `Orchestrator` sends the preprocessed query to the `Equivalence Handler` (step 5) in order to verify if any of the preprocessed query terms is present in a synonym node. In this case, the word *Quicksort* was found as synonym of the main node *ordenamiento rapido*, for that reason this word was added as an expansion term. The relevance of this term ends here because the word Quicksort is not ambiguous. On the other hand, the word *pila* also goes through the mentioned mechanisms, without major relevance in respect to the `Equivalence Handler`. Continuing with the flow, the `Ambiguity Handler` (step 6) calculated the similarity values between the ambiguous token and the supplementary one. In this case, the `AplicacionesPilas` and

TADPilas nodes reached the same optimum because both of them include the term *pila*; however, inside the ontology both refer to different topics related to stacks. In this point the user information gains relevance for the disambiguation because in the ontology structure the AplicacionesPilas node belongs, by transitivity, to the Algoritmia (AL) course, which includes training in problems and applications of stacks, and the TADPilas node belong to the Fundamentos de Programacion (Programming Fundamentals - PF) course, which includes theoretical instruction on the Stack abstract data type. When previously consulted to the DB, it was determined the user is currently taking the AL course, but not PF. For that reason, the node from the AL course was selected as a better match. The final expanded query is *pila ordenamiento rapido aplicacion quicksort* which is sent to the IR Engine (step 7) in charge of retrieving the information and outputting the ranked list of documents (step 8).

When using the IR engine without QE, 6 documents were retrieved, all of them containing the word *pila* in their tags, and the ones in the top were mostly related to the theory of the abstract data type of the PF course. From the 6 documents, just two of them were relevant, which led to a 33% of precision. When using QE 10 documents were retrieved. The one in the top was indeed the most relevant which included in the same university examination exercises about applications of stacks and the quicksort method, and gradually other documents relevant to a lesser degree. From the 10 documents, 7 of them had a degree of relevance, which led to a 70% of precision.

VI. Discussion

After the execution of tests we realized some stopwords would better be excluded from the stopwords list. This is the case of the word *no* from the query *no programacion en pascal* (not pascal programming). Even though the word *no* can be considered a stopword, its existence in order to keep the semantic integrity of the complete query is considered a relevant factor to take into account. However, for this work, those particular scenarios will not affect the results because the scope of this model excludes the analysis by propositional logic. We use the user information only for disambiguation cases, and don't directly include that information as terms for the expansion. That is because the information obtained regarding the user may be more general, and adding it as terms for the expansion in the total of cases could generalize the query and negatively affect the precision measure.

VII. Conclusion and future works

The developed tool based on the proposed model has proven that from an ambiguous query was possible to retrieve relevant information for the user. Different tests were performed in order to measure the tool. Tests without using QE resulted in an overall of 16.5% of precision, whereas tests using QE resulted in an overall of 69% of precision, so it reaffirms the proposed model led to better results. From a point of view of benefits, the selected domain is particularly useful for academic purposes, because the tool can be used for students from careers related to CS to retrieve relevant information for their studies or research projects. In second place, this model is generic, so if the ontology is changed to another which follows the described structure, it is possible to perform the searching. In third place, the result of this work is a conceptual model which can be implemented on different platforms and without dependencies on specific technologies.

It would be interesting to test the model with other different domain ontologies. Regarding the mechanisms, it would be very useful one that pulls information from online sources and another that captures in an automatic way the always dynamic user information. Finally, we propose the inclusion of further query preprocessing which takes into account propositional logic, and so stopwords dictionary can consider those cases for accurate results.

References

[1] P. H. Cleverley and S. Burnett, "Retrieving haystacks: a data driven information needs model for faceted search," *Journal of Information Science*, vol. 41, no. 1, pp. 97–113, 2015.

[2] M. C. de Andrade and A. A. Baptista, "Researchers' information needs in the bibliographic database: A literature review," *Information Services and Use*, vol. 34, no. 3, pp. 241–248, 2014.

[3] Y. Gupta, A. Saini, and A. K. Saxena, "A new fuzzy logic based ranking function for efficient information retrieval system," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1223–1234, 2015.

[4] M. Mitra and B. B. Chaudhuri, "Information retrieval from documents: A survey," *Information Retrieval*, vol. 2, no. 2-3, pp. 141–163, May 2000.

[5] F. Ren and D. B. Bracewell, "Advanced information retrieval," *Electronic Notes in Theoretical Computer Science*, vol. 225, no. 0, pp. 303–317, 2009.

[6] M. Kobayashi and K. Takeda, "Information retrieval on the web," *ACM Comput. Surv.*, vol. 32, no. 2, pp. 144–173, 2000.

[7] M. Mitra, A. Singhal, and C. Buckley, "Improving automatic query expansion," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '98. New York, NY, USA: ACM, 1998, pp. 206–214.

[8] G. J. Hahm, M. Y. Yi, J. H. Lee, and H. W. Suh, "A personalized query expansion approach for engineering document retrieval," *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 344–359, 2014.

[9] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International Journal of Human-Computer Studies*, vol. 43, no. 56, pp. 907–928, Nov. 1995.

[10] H. Wang, Y. Guo, X. Shi, and F. Yang, "Conceptual representing of documents and query expansion based on ontology," in *Web Information Systems and Mining*, ser. Lecture Notes in Computer Science, F. L. Wang, J. Lei, Z. Gong, and X. Luo, Eds. Springer Berlin Heidelberg, Jan. 2012, no. 7529, pp. 489–496.

[11] H. Melgar S., D. Salas Guillen, and J. Gonzales Maceda, "Ontology based inferences engine for veterinary diagnosis," in *Semantic Technology*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 79–86.

[12] S. Kara, . Alan, O. Sabuncu, S. Akpnar, N. K. Cicekli, and F. N. Alpaslan, "An ontology-based retrieval system using semantic indexing," *Information Systems*, vol. 37, no. 4, pp. 294–305, 2012.

[13] M. Fernandez, I. Cantador, V. Lpez, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced information retrieval: An ontology-based approach," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 4, pp. 434–452, 2011.

[14] V. Jalali and M. Borujerdi, "The effect of using domain specific ontologies in query expansion in medical field," in *International Conference on Innovations in Information Technology, 2008. IIT 2008*, Dec. 2008, pp. 277–281.

[15] W. Ali and S. Khan, "Ontology driven query expansion in data integration," in *Fourth International Conference on Semantics, Knowledge and Grid, 2008. SKG '08*, Dec. 2008, pp. 57–63.

[16] Z. Lu, Z. Yan, and L. He, "OnPerDis: Ontology-based personal name disambiguation on the web," in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 1, Nov. 2013, pp. 185–192.