Description Logics-Based Modelling for Precise Information Retrieval

Saïd Radhouani and Gilles Falquet

Centre Universitaire d'Informatique
University of Geneva, Genève, Switzerland
{radhouani, falquet}@cui.unige.ch

1 Introduction

In professional environments, users have a good knowledge about their domain of interest as well as the documents[1] they consult regularly. In order to carry out their professional tasks[2], they need an Information Retrieval System (IRS) that allows them to find a precise answer to their information needs. Generally speaking, they know about documents content that may satisfy their information needs. Thus, during the retrieval task, they try to complete the information that they have and that is insufficient. Their information needs are in this case formulated through **precise queries**.

The qualifier "precise" denotes a query that contains: *i*) **a very specialised terminology** and *ii*) **a complex structure**. Through a precise query, a user can describe his information need using explicit semantic relationships between the descriptors of his query. He also can use boolean operators or quantification (at least, all, etc.) in order to specify the number of elements that the desired document should contain. In order to illustrate some characteristics of precise queries, we present here some query examples.

Query 1 "*Give me documents that deal with the French football player who got a red card during the final of the FIFA 2006 world cup*".

Through this query, the user is looking for a *football player* whose *nationality* is *France*. A relevant document can for instance contain the name "Zinedine Zidane" without necessarily containing the terms "football player" and "French". Such a document cannot be found by a system based on term matching. A possible solution is to specify that "football player" and "French" are not the terms the user is looking for, but rather *descriptions* of the elements of interest. In this case, the system needs some additional knowledge to infer that Zinedine Zidane is a French football player.

Query 2 "*Give me images with a tibia without any pathology*".

[1] Medical report, law text, etc.
[2] A diagnosis, write a newspaper article, etc.

The user is looking for images containing a *tibia* without any *pathology* (no fracture, no dislocation, etc.). A relevant document must therefore contain the *tibia* and must not contain any *pathology* affecting it.

It is possible that a relevant document contains a tibia without pathology and other parts of the human anatomy affected by other pathologies. For this reason, we must distinguish, during the retrieval process, that only documents containing tibia affected by pathologies are to be excluded. This can be expressed by using a relationship *affected_by* between the query descriptors: *tibia* and *pathology*.

Query 3 "*Give me documents dealing with Bill Gates and Steve Jobs and at least two computer companies*".

The user is looking for a document dealing with two persons whose names are known: *Bill Gates* and *Steve Jobs*, and at least two organisations described by a generic description: *computer companies*. Thus, a document dealing with two persons and less than two computer companies cannot be considered as a relevant answer. In order to allow the IRS to correctly interpret this information need, it is necessary to provide a quantification operator that allows the user to specify the number of elements he is looking for.

Through these examples, it is clear that solving precise queries requires, during the indexing and the querying processes, to take into account:

1. the very specialised terminology of the document (query) descriptors;
2. the relationships between the descriptors;
3. the complex structure of the query.

For items 1 and 2, a possible solution is to use specialised domain knowledge, which can be modelled through an **external**[3] **resource**[4] (ER). Indeed, an ER can be used to extract the specialised vocabulary and therefore highlight the relevant elements that contribute to the description of the document (query) semantic content. The semantic relationships modelled within the ER can be used between the descriptors during the document (query) description. Relationships can also be used during the retrieval process. For example, the relation *is-a* can be used to return a document containing "Zinedine Zidane" for a query asking "French football player".

In order to produce a precise description of the document's semantic content, we need an expressive document language, which allows to use the specialised terminology and the relationships modelled within the ER.

For item 3, we need an expressive query language, which allows the user to use: *i*) the very specialised terminology; *ii*) the relationships between the descriptors of his query; and *iii*) the desired operators.

This paper is organised as follows: In Section 2, we will present the most significant approaches that use ER for information retrieval (IR). Section 3 will

[3] "external" because it models knowledge, which are not present in the documents (queres) to be processed, at least in an explicit and complete form.

[4] Ontologies, thesaurus, taxonomies, etc.

be dedicated to the knowledge formalism we chose for our modelling. In Section 4, we will define our IR model presenting the document model and the query model in detail. Before concluding (Section 6), we will present some experiments and discuss them (Section 5).

2 Related work

There are mainly two categories of approaches that use ERs for IR: the *conceptual indexing*[1][2][3] and the *query expansion* [4][5][6]. Both of them require a disambiguation step, which allows to identify, from the ER, the concepts denoted by the words within the document and the query [7][8].

The conceptual indexing consists in representing documents (queries) by concepts instead of ambiguous words [9][10][11]. Thus, during the retrieval process, the matching between a query and a document is done based on non-ambiguous vocabulary (concepts). So far, the approaches based on this technique have not given significant improvement in terms of retrieval performance [9][12]. One of the factors on which depends the retrieval performance is the method used to "interpret" the semantic content of the document (query). In existing approaches, once the concepts are extracted, the documents (queries) are considered as "bags of concepts". Therefore, the semantic relationships that may exist between the concepts they contain are not exploited. Consequently, the documents dealing with a subject close to that of the query could not be found with these approaches. Some works have shown interest in the representation of documents by semantic networks that connect the concepts of the same document. However, these networks are used only for disambiguation and not during the IR process [9]. The query expansion is a possible solution to this problem [5][6][13].

The idea behind query expansion is to use semantic relationships in order to enrich the query content by adding, from the ER, concepts that are semantically related to those of the query [5][6][13][14]. Several works analysed this aspect, but few have had positive results. In response to these failures, researchers proposed to extend the queries in a "careful" manner by selecting some specific relationships during the expansion process [4][9]. This improves the retrieval performance [9], but, the extended queries are still considered as bags of concepts, and their structure is ignored during the retrieval process.

The existing approaches seem to be insufficient considering the requirements that we have presented. Indeed, they treat documents and queries as bags of concepts and do not enough consider their structure. They are therefore incapable to solve precise queries which have a complex structure and require a specific treatment in order to highlight all aspects related to their semantic content.

We believe that in addition to describing the domain knowledge, ERs can provide useful information for interpreting the semantic content of the documents (queries). In this direction, the approach we have adopted consists in using concepts and relationships in order to add a semantic structure to the representation of documents (queries). Concepts and relationships are used for: *i*) extracting the relevant elements that contribute to the description of the se-

mantic content of the document (query), and *ii*) matching query and document even if they do not share the same words.

In order to incorporate domain knowledge during the definition of our IR model, it is suitable to have a uniform representation of documents, queries, and the ER. This can be reached using an appropriate **knowledge representation formalism**, which is common to these three elements. This formalism must: *i*) allow the manipulation of all knowledge treated by our system; *ii*) take into account user requirements in terms of operators; and *iii*) offer a comparison operation that can implement the matching function of our IRS.

3 Formalism for knowledge representation

Several formalisms have been used in the IR modelling, notably Semantic Trees [16], conceptual Graphs [17] and Description Logics (DLs) [18]. Taking into account our needs, we found out that DLs are particularly appropriate for modelling in our context. Indeed, DLs allow to represent the three sources of knowledge (documents, queries and ER) by the same formalism, which ensures that all these sources can participate in the IR process in a uniform and effective way. This formalism provides also a high level of expressiveness, which is very suitable for the representation of precise information needs. For example, it contains all the operators we need in our model. It also allows to use concepts and relationships during the modelling process. And as we will show in the next Sections, it offers a comparison operation that can implement the matching function of the IRS. According to our needs, we chose the $\mathcal{ALCQ}$ DL language.

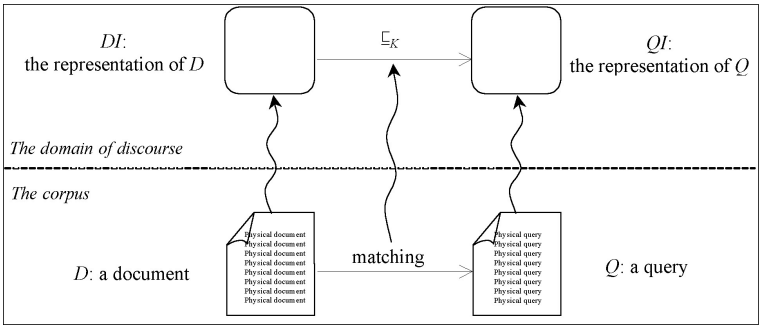

Fig. 1. Matching between a query and a document represented in DL.

The application of the Description Logic in the field of IR is immediate because it is sufficient to consider the documents collection as a subset of the chosen domain of discourse and to represent the documents and queries by concepts. Thus, each document D (query Q) will be represented in the knowledge base K by its index DI (QI), which is an $\mathcal{ALCQ}$ expression. DI is an abstraction

(representation) of a set of documents that have the same content. Thus, the physical documents correspond to the instances of DI. According to the DLs terminology, the correspondence between a query Q and a document D is calculated within the subsumption hierarchy: *the document D is relevant for the query Q if the concept DI is subsumed by the concept QI*: $DI \sqsubseteq_K QI$ (figure 1).

4 Information retrieval model for precise queries

In this Section, we present our IR model's specificities compared to the existing approaches, and we define its components, namely the knowledge base model, the document model and the query model.

4.1 Knowledge base model

We present here the formal model of the knowledge base K describing the domain knowledge present in the corpus.

Let $C = \{c_1 \dots c_{nc}\}$ a set of nc atomic concepts, $R = \{r_1 \dots r_{nr}\}$ a set of nr roles, and $S = (C, R)$ the signature of K. Once the signature S is fixed, an interpretation I for S is a pair $I = (\Delta^I, .^I)$, where:

- Δ^I is a non-empty set;
- $.^I$ is a function assigning:
  - a subset $C_i^I \subseteq \Delta^I$ for each atomic concept $c_i \in C$;
  - a relation $R_i^I \subseteq \Delta^I \times \Delta^I$ for each role $R_i \in R$.

Given the DL language $\mathcal{ALCQ}$ and a signature S, a knowledge base K within $\mathcal{ALCQ}$ is a triple $K = (S, T, A)$, where T, a Terminological Box (TBox), contains a set of terminological axioms in the form: $C \equiv D$, or $C \sqsubseteq D$, C and D are two $\mathcal{ALCQ}$ expressions on the signature S, and A is the Assertion Box (ABox).

The empty ABox In our case, the *ABox* is empty because we represent documents and queries by concepts. So, for instance, the term *Jacques*, which would usually represent an instance of the concept *Person*, will give rise to a concept *Jacques* $\sqsubseteq$ *Person*. We have been led to this choice mainly because in many available external resources (thesauri, lexical ontologies, taxonomies) there is no clear distinction between instances and concepts. In addition, we are not in a database querying context where queries refer exclusively to instances, here queries contain may refer to 'instances" (*Stave Jobs*) and "concepts" (*computer*). Considering everything as a concept offers a unified framework.

4.2 Specificities of our model

During the user's query formulation, we can distinguish two non exclusive scenarios:

1. The user knows the concept identifying the element he is looking for (eg. the name of a *person*: Angela Merkel, a name of a part of the *human anatomy*: femur, etc.). In this case, the desired element is called **identified** by the user.
2. The user knows some properties that can describe the element he or she is interested in. For example, the element's *occupation* is *chancellor*, and its *nationality* is *German*. Thus, the desired element is not identified but **described** by its relationships to other concepts.

In order to take into account both scenarios, it is necessary to identify these two kinds of elements during the indexing and the querying processes. For this purpose, we propose a new indexing unit: the *semantic descriptor*.

Definition: A semantic descriptor within a document (query) is defined by a set of concepts and semantic relationships. Any concept from the knowledge base can define a semantic descriptor. According to the previous scenarios, a semantic descriptor can be **identified** or **described**.

In order to define the semantic descriptors within the description of a document (or query), we propose to use the following relationships:

identified_by: we use this relationship to define a semantic descriptor by connecting it with the concept that allows to identify it.

described_by: we use this relationship to describe a semantic descriptor by connecting it to those concepts that describe it. The name *described_by* represents a very generic relationship; in practical applications it will be replaced by a relationship (role) of the knowledge base. For example, in order to describe the semantic descriptor *person* by the relationship that connects a person to his or her country of origin we would use the role *country_of_origin*.

Formally, a semantic descriptor S is an $\mathcal{ALCQ}$ expression of the form:

$$S \equiv d_{idf} \sqcap \exists\, described\_by\,.\,c_1 \sqcap \ldots \sqcap \exists\, described\_by\,.\,c_n$$

where:

- d_{idf} is the concept that allows to identify S,
- each c_j is a concept (atomic or defined by an expression).

Example: In a document containing *the French football player Zidane*, the semantic descriptor corresponding to this person is *identified by Zidane* and *described by football player* and *France*. Formally, this semantic descriptor corresponds to the concept definition

$$S \equiv Zidane \sqcap \exists occupation\,.\,football\_player \sqcap \exists nationality\,.\,France$$

4.3 Document model

Each document *doc* is represented by a concept I_{doc} defined by the conjunction of the semantic descriptors belonging to *doc*. In order to represent the documents and the queries using semantic descriptors, we propose to use the role

indexed_by, which allows to associate a semantic descriptor S to a given document *doc*. Formally, the index I_{doc} of a given document *doc* containing the semantic descriptors $\{S_1, \ldots, S_n\}$ is the concept

$$I_{doc} \equiv \exists\, indexed\_by.S_1 \sqcap \ldots \sqcap \exists\, indexed\_by.S_n$$

After the indexing process, the index of the document collection is comprised of the original TBox extended by the I_{doc} concepts.

4.4 Query model

Each query q is represented by a concept I_q, which is defined using the atomic concepts, the role used in the semantic descriptors and the *indexed_by* role.

During the querying process, the TBox is extended with the concept I_q.

Example: A relevant document to *Query 2* must contain a tibia *without any* pathology affecting precisely this part of the human anatomy. Thus *Query 2* is represented in our model by the following expresion:

$$I_{Q2} \equiv \exists\, indexed\_by.(Tibia \sqcap \neg\exists\, affected\_by.Pathology)$$

Example: *Query 3* refers to three semantic descriptors : *Bill_Gates*, *Steve_Jobs*, and *computer_company*. It is represented by the concept

$$I_{Q3} \equiv \exists\, indexed\_by.Bill\_Gates \sqcap \exists\, indexed\_by.Steve\_Jobs \sqcap \geq 2\, indexed\_by.computer\_company$$

Example: *Query 4* "Give me an image containing Martin Luther King **alone**". Thus, a relevant document must contain **one and only one** *Person*: *Martin Luther King*. The corresponding expression is

$$I_{Q4} \equiv \exists\, indexed\_by\,.\, Martin\_Luther\_King \sqcap \leq 1\, indexed\_by\,.\, Person$$

Retrieval process: The retrieval process consists in selecting the documents that satisfy the query requirements from the indexed documents. In DL terms, the retrieval process can be seen as a task to retrieve those documents represented by concepts that are subsumed by the concept representing the corresponding query. Thus, the matching between a query Q and a document *doc* is done by verifying that $I_{doc} \sqsubseteq I_Q$ is true within the knowledge base K. Finally, the set of relevant documents for a given query Q is $\{doc \,|\, I_{doc} \sqsubseteq_K I_Q\}$

5 Experimental implementation

The implementation of our model requires the following steps:

1. finding or creating an external resource that is suitable for the application domain

2. creating the initial TBox, in a suitable DL language, from the external resource
3. extracting the semantic descriptors from documents;
4. representing documents using the extracted semantic descriptors and complementing the TBox with these descriptions;
5. retrieving relevant documents for a given query with a reasoner for the chosen DL language.

For our experiments, we chose to work on the medical domain, which is a typical environment where user queries are often precise. Thus, we used the ImageCLEFmed collection, which contains medical reports (texts and images). This collection was used during the CLEF-2005 IRS evaluation campaign [19]. For our experiment needs, we selected parts of the UMLS[5] metathesaurus as external resource. The selected parts were the subhierarchies of concepts having as roots *Anatomical part*, *Pathology*, and *Image modality*. The reason for this choice was, we observed that most of the precise queries in the medical imaging domain refer to concepts in at least one of these hierarchies.

We developped a tool which allows to represent the selected UMLS part in OWL to crete the initial TBox [20]. Then we extended the TBox with the document descriptions. The description extraction was limited to

1. the recognition of terms denoting concepts of the initial TBox (for the *identifiy_by* part of the descriptors)
2. pattern matching to recognize a few relationships (for the *described_by* part of the descriptors) .

Once the TBox is constructed, the third step can be realised using any existing DL reasoner (Racer, Fact++, Pellet, etc.). For query answering, the reasoner carries out the inference in the TBox and computes the subsumption hierarchy. Relevant documents for a query Q are those that are represented by the concepts which are subsumed in the computed hierarchy by the concept I_Q.

We have conducted several experiences on non-trivial queries from the imageCLEFmed collection. For a given query, we compare the result given by our system to those given by a baseline system based on the vector space model [21]. The obtained results are very promising and overperform those obtained by the baseline system, which considers documents and queries as bags of concepts.

Discussion: We conclude that the design of the used ER has a major impact on search result. Indeed, the matching function based on the calculation of the subsumption can be very beneficial when the ER is rich in terms of *is-a* relationship. Indeed, through the algorithm that computes the subsumption, the use of DL offers a capacity of reasoning that can deduce the implicit knowledge from those given explicitly in the TBox, and therefore help to retrieve relevant documents for a given query even if they do not share any words with it.

[5] *Unified Medical Language System* [http://www.nlm.nih.gov/research/umls/ain-text]

However, we encountered some problems using the subsumption hierarchy. Indeed, depending on the domain, the ER may be organized according to different semantic hierarchies. For instance, in the geographic domain, the geometric containment is probably one of the most important hierarchical relationship. The same is true for human anatomy. For example, if a user looks for a *fracture in the leg*, he or she will certainly consider a document dealing with a *pathology of the tibia* as relevant. Thus the retrieval system must take into account the *part_of* hierarchy that exists within the human anatomy. One way to solve this problem is to twist the subsumption relation and to represent the *part_of* hierarchy as a subsumption hierarchy. Thus implicitly stating, for instance, that *a tibia is a leg*.

In this approach, a query

$$I_Q \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,Leg)$$

will correctly retrieve a document described by

$$I_D \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,Tibia)$$

because $I_D \sqsubseteq I_Q$ if $Tibia \sqsubseteq Leg$. We have implemented this "quick and dirty" approach in our early experiments. However, using subsumption to mimic another relation may lead, in certain circumstances, to unexpected and conter-intuitive deductions. A "cleaner" and semantically safer approach consists in defining transitive properties to represent the various types of hierarchies that may exist in a given domain. The above example would then lead to the following descriptors:

$$I_Q \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,(\exists part\_of.Leg)$$

$$I_D \equiv \exists indexed\_by\,.\,(Fracture \sqcap \exists\, location\,.\,Tibia)$$

If an axiom specifies that *part_of* is transitive and the definition of *Tibia* is of the form "... $\sqcap\ \exists part\_of.Leg$" then the reasoner will infer that $I_D \sqsubseteq I_Q$. Although semantically sounder, this requires a slightly more complex query formulation.

Performance considerations: It is obvious that using DL reasoners to perform IR tasks leads to performances that are several orders of magnitude slower than classical index-based IRS. Nevertheless, several points could be worth studying to improve the DL approach performances: *i*) document descriptors are generally simple (limited to $\sqcap$ and $\exists$ constructors) thus we could devise simpler reasoning algorithms, *ii*) when queries are simple, reasoning becomes even simpler and *iii*) the document corpus is generally stable and could be pre-processed in some way to facilitate the reasoner's work.

6 Conclusion

We presented an information retrieval model able to solve precise queries. It is based on a new indexing unit defined by concepts and relationships: the *semantic*

descriptor. The indexing and querying processes are supported by an external resource describing the user's interest domain knowledge. Thus, using expressive languages, we represent documents and queries by semantic descriptors.

In order to define our model, we chose the *Description Logic*, which allows a uniform precise representation of documents and queries semantic content. Thus, we showed how a document model, a query model and an external resource can be adapted within a DL in order to participate in the IR process in a uniform and effective way. We also presented how the subsumption can be used as a matching function allowing to implement the system's relevance.

The results obtained during the experimental implementation of our model confirmed that the use of DL has a very good impact on the retrieval performance. Indeed, DL offers the opportunity to use background knowledge about a specific domain. Thus, during the querying process we can benefit from the powerful reasoning capabilities a reasoner offers, notably the capacity to deduce the implicit knowledge from knowledge given explicitly in the TBox.

References

1. Biemann, C.: Semantic indexing with typed terms using rapid annotation. In: Proceedings of the TKE-05-Workshop on Methods and Applications of Semantic Indexing, Copenhagen (2005)
2. Mihalcea, R., Moldovan, D.: Semantic indexing using wordnet senses. In: Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval, Morristown, NJ, USA, Association for Computational Linguistics (2000) 35–45
3. Vallet, D., Fernández, M., Castells, P.: An ontology-based information retrieval model. In Gόmez-Pérez, A., Euzenat, J., eds.: ESWC. Volume 3532 of Lecture Notes in Computer Science., Springer (2005) 455–470
4. Qiu, Y., Frei, H.P.: Concept based query expansion. In Korfhage, R., Rasmussen, E.M., Willett, P., eds.: SIGIR, ACM (1993) 160–169
5. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, Springer-Verlag New York, Inc. (1994) 61–69
6. Baziz, M., Aussenac-Gilles, N., Boughanem, M.: Désambiguisation et Expansion de Requêtes dans un SRI, Etude de l'apport des liens sêmantiques. Revue des Sciences et Technologies de l'Information (RSTI) série ISI **8** (2003) 113–136
7. Krovetz, R., Croft, W.B.: Lexical ambiguity and information retrieval. ACM Transactions on Information Systems **10** (1992) 115–141
8. Sanderson, M.: Word Sense Disambiguation and Information Retrieval. Ph.d. thesis, University of Glasgow, Glasgow G12 8QQ, UK (1997)
9. Baziz, M.: Indexation conceptuelle guidée par ontologie pour la recherche d'information. Thèse de doctorat, Université Paul Sabatier, Toulouse, France (2005)
10. Smeaton, A., Quigley, I.: Experiments on using semantic distances between words in image caption retrieval. In: Proc. of 19th International Conference on Research and Development in Information Retrieval, Zurich, Switzerland (1996)

11. özlem Uzuner, Katz, B., Yuret, D.: Word sense disambiguation for information retrieval. In: AAAI/IAAI. (1999) 985
12. Voorhees, E.M.: Natural language processing and information retrieval. In Pazienza, M.T., ed.: SCIE. Volume 1714 of Lecture Notes in Computer Science., Springer (1999) 32–48
13. Mihalcea, R., Moldovan, D.I.: An iterative approach to word sense disambiguation. In: Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press (2000) 219–223
14. Baziz, M., Boughanem, M., Aussenac-Gilles, N., Chrisment, C.: Semantic cores for representing documents in ir. In: SAC '05: Proceedings of the 2005 ACM symposium on Applied computing, New York, NY, USA, ACM (2005) 1011–1017
15. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
16. Berrut, C.: Une méthode d'indexation fondée sur l'analyse sémantique de documents spécialisés. Le prototype RIME et son application à un corpus médical. Thèse de doctorat, Universitè Joseph Fourier, Grenoble, France (1988)
17. Chevallet, J.P.: Un Modèle Logique de Recherche d'Informations appliqué au formalisme des Graphes Conceptuels. Le prototype ELEN et son expérimentation sur un corpus de composants logiciels. PhD thesis, Université Joseph Fourier, Grenoble (1992)
18. Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (1993) 298–307
19. Clough, P., Muller, H.: The clef cross language image retrieval track 2005. In: http://ir.shef.ac.uk/imageclef2005/. (visited on November 2005)
20. Kashyap, V., Borgida, A.: Representing the umls semantic network using owl: (or "what's in a semantic web link?"). In Fensel, D., Sycara, K.P., Mylopoulos, J., eds.: International Semantic Web Conference. Volume 2870 of Lecture Notes in Computer Science., Springer (2003) 1–16
21. Salton, M.J.M.G.: Introduction to modern information retrieval. McGraw-Hill, New York (1983)