Query Ranking in Information Integration

Rodolfo Stecher, Stefania Costache, Claudia Niederée, and Wolfgang Nejdl

L3S Research Center, Appelstr. 9a, 30167 Hannover, Germany
{stecher,costache,niederee,nejdl}@L3S.de

Abstract. Given the growing number of structured and evolving online repositories, the need for lightweight information integration has increased in the past years. We have developed an integration approach which relies on partial mappings for query rewriting and combines them with a controlled way of relaxation. In this paper we propose a novel approach for ranking results of such rewritten and relaxed queries over different sources, by punishing lack of confidence in mappings used for rewriting, as well as punishing higher degrees of controlled relaxation introduced, and present the performed evaluation.

Keywords: Query Ranking, Information Systems, Query Relaxation.

1 Introduction

A growing number of structured large information collections is made accessible over the Internet, e.g. Freebase, Linked Data, also in the area of Personal Information Management, new repositories as Flickr, YouTube are increasingly used. Efforts in adding structure and giving semantics to the available information, result into further information collections (e.g. UMBEL, DBPedia). The easy exploitation of such sources, however, requires approaches for flexible, lightweight integration of distributed and evolving structured information sources. Thus, lightweight approaches are required for query answering over evolving information systems with reasonable result quality, even when only partial mapping information is available (incomplete mappings with various confidences). We follow a lightweight approach similar to a pay-as-you-go (results are as good as possible given the available evidences) integration, which uses partial mappings for rewriting and relaxing structured (triple-based) queries and learning of new mappings from the results, as described in more detail in [1].

Even with such a solution at hand, the multitude of data within information systems makes finding the needed results still difficult, since relaxed queries do not always provide an exact set of results, but rather an extended, more permissive set of results. When information comes from various sources with different degrees of confidence, an ordering among the results reflecting this confidence is needed. In this paper we propose a ranking algorithm which punishes lack of confidence in the used partial mappings and query rewriting strategies.

The main contributions of our approach can be summarized as follows: 1) An innovative query ranking approach for our lightweight information integration

B. Pernici (Ed.): CAiSE 2010, LNCS 6051, pp. 230–235, 2010.

approach [1]; 2) An evaluation of the proposed ranking approach using large real world data sets showing its applicability in realistic settings.

Next, Section 2 presents our query ranking approach, its evaluation is presented in Section 3, Section 4 presents the related work, and Section 5 summarizes our conclusions and future work.

2 Ranking for Lightweight Information Integration

After applying our relaxation strategies [1], we use the ranking of *queries* to rank results from different information sources. Considering that complete and confident mappings can only give correct results (the ideally rewritten query), our approach for a *query* ranking function is based on introducing punishments to each of the aspects which might introduce errors. Intuitively, the higher the difference of the resulting query from the ideally rewritten query, the more "punishment" we add to the results obtained from executing it. Due to space constraints we refer to the definitions presented in [1].

2.1 Punishment Derived from Mappings

The higher the confidence in the correctness of a mapping is, the lower the probability of introducing an error when this mapping is used to rewrite the user query. For reflecting this, and also considering that mappings are independent, we define a factor called **Query Mapping Confidence** (Qmc), reflecting the probability of introducing errors by the product of the confidences of all mappings actually used in the rewriting of Q^u, multiplied once (as also done in [2]):

$$Qmc = \prod v | (e, e', v) \in M_i' \tag{1}$$

The rationale for this is that each successfully applied mapping will avoid that this element is relaxed with a wildcard, but it will also possibly add some incorrect results, depending on the confidence of this mapping.

2.2 Punishment Derived from Relaxation

Our query relaxation might introduce errors by allowing bound expressions in the user query Q^u to become unbound in the reformulated query Q^{S_i}. The computation of the punishment considers the number of introduced variables which remain bound and the number of variables introduced which do not even have a value constraint. The computation relies on the following additional notations:

- qL: the number of triple elements in Q^u: qL = Number of triples $*3$
- $\mathcal{LU}$: the element names in Q^u not part of $A^u \cup \tau \cup VAR_{Q^u}$ (i.e. Literals and URI's not belonging to the schema)
- nb and nu: the number of bound and unbound variables introduced in the relaxation process (defined below)

Next we define the introduced number of bound and unbound variables:

- nb: For $< s, p, o >$, p is replaced with var_p: $< s, var_p, o > \wedge o \in \mathcal{LU}$, then $nb = nb + 1$ once for var_p. This is, nb will be incremented by one the first time var_p is introduced, even though any other occurrence of p will also be replaced with the same var_p. The idea is that we are introducing a wildcard, but the values it can take are restricted by s and the given value of o which is fixed.
- nu: we have two options: 1) For $< s, \tau, o >$, o is replaced with var_c: $< s, \tau, var_c > \wedge o \in A_C$, $nu = nu + 1$ for *every* usage of var_c. We increase nu for every occurrence, since more relaxation is added with every replacement. In this case we are allowing s to be of any type, as well as any other thing originally specified to have the same type as s, so we are relaxing the "essence" specification of things; and 2) For $< s, p, var_o >$, p is replaced with var_p: $< s, var_p, var_o > \wedge var_o \in VAR_{Q^u}$, $nu = nu + 1$ for *every* occurrence of var_p. We increase nu since each occurrence increases the degree of relaxation because we had already only the relation p as a restriction between the values that s and var_o could get, and now we are even relaxing this last restriction. In this case the relaxed query expresses the fact that there must be some connection between s and var_o, but without saying which connection.

With these measures, we can compute a punishment for bound variables as:

$$P_{nb} = 1 - \frac{nb}{qL} \tag{2}$$

We need this factor, since even though the variables are bound, their number influences the relaxation relative to the query joins (therefore the normalization to the query length). Also, the correlation between nb and the ranking is indirect, since the less bound variables are introduced, the higher we can rank that query. The introduction of unbound variables has a higher influence on the accurateness of the query results than the bounded ones, because they introduce more relaxation to the query, and therefore more penalty is needed. We define the punishment for unbound variables as:

$$P_{nu} = \alpha^{1-\sqrt{\frac{nu}{qL}}} - 1, \tag{3}$$

where α, the relaxation penalty, has to be defined. It grows differently than the one computed for bound variables: errors introduced by unbound variables make the query less accurate than introduction of bound variables, and therefore this factor should have a more dramatic decrease when the number of unbound variables is high. We experimented with different functions for the exponent factor of P_{nu}, but they all behaved similarly.

2.3 Ranking Function

Based on the factors presented above, we define a ranking function, giving higher ranking to results of queries with likely less errors. The factors presented in Equations 1, 2 and 3 are probabilistically independent and therefore the ranking function can be computed as a product of the possible punishments (errors) introduced:

$$R(Q^u) = Qmc * P_{nb} * P_{nu} \tag{4}$$

$R(Q^u)$ estimates the expected correctness of the modified query results, by considering the confidence of the mappings, and the effects of applying a "wildcard" strategy on the original query. The combination of these values leads to a measure of how much the modified query "deviates" from the ideally rewritten query.

3 Evaluation

By computing the precision for different top-k results, we prove the ranking method to be efficient in ordering the results. Since for the ideal query, all results are equally correct, we could not compare our results against a ranked ground truth, instead, we used the complete ground truth. As strategies $\mathcal{SUB}^2$ and $\mathcal{SUB}^4$ presented similar behavior as $\mathcal{SUB}^6$ they will not be discussed in detail.

3.1 Evaluation Setting

Information Sources. The heterogeneous information sources considered with the number of contained triples are presented in Table 1. In detail they are:

Virtual Personal Desktop (VPD) obtained from crawling 16 desktops (PDFs, Word documents, Emails, Wiki pages, FOAF profiles) using the crawling approach and ontologies presented in [1], each with their own user ontology and query set (see [1] for details).
UMBEL (`http://www.umbel.org/`) provides an ontology and its instances, along with many definitions of equality (using "sameAs" relations) to instances in the other datasets used for this experiment - YAGO and DBPedia. Therefore, the instances provided by UMBEL were used to compute the ground truth. The *UMBEL Ontology* was taken as the *user ontology* (in a filtered version), and will be denoted from now on as user ontology O^u. In order to construct the *UMBELInstances* data source and its ontology, original references to concepts and properties, as well as to resources in the instances were modified programmatically, simulating in this way a new source, from now on UMBEL.

YAGO [3] - from the provided instances a simple ontology (*YAGO Ontology*) describing them was extracted.

DBPedia (`http://wiki.dbpedia.org/`) - two sources were created: the *DBPediaPersons* containing all available "Persondata" files which are represented using the *FOAF* ontology, and the *DBPediaInfoboxes* containing the "Infoboxes" and "Types" files which are represented using the *DBPedia Ontology*.

Table 1. Information Sources Overview

Source	VPD	UMBEL	YAGO	DBPediaPersons	DBPediaInfoboxes
No. triples	3,329,376	6,740,428	460,418,591	812,339	9,877,564

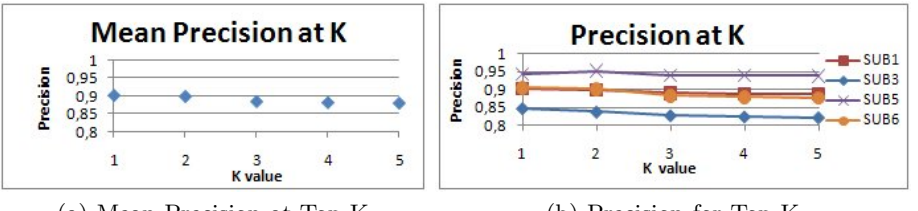
Initial Mappings. Were computed between the *UMBEL Ontology* and the ontologies of the sources using [4], and later randomly modified to serve as initial *partial* mappings. Partial mappings were also computed for the VPDs, between the user ontology specified in [1] and the ontologies describing the different crawled sources.

Queries. Our query set is an extension of the queries used in [1] and consists of more than 70 queries containing mainly 1 to 3 joins.

Ground Truth. For each query there is a ground truth, which contains all correct results expected. In order to have comparable query results from the different sources, explicit equivalences between resources have been exploited.

3.2 Experiments

For each information source, we employ relaxation strategies (wildcard based) together with rewriting strategies (mapping based), and the techniques for simulating user feedback and learning of mappings as described in [1]. We compared the top-k *integrated* results of the same query over all available information sources with the ground truth and measured the precision at top-k (top-1 to top-5) using a precomputed value of $\alpha = 2$ (the experiments for determining the α value are not presented due to space constraints). The mean results obtained from running this evaluation over all presented datasets, iterations and strategies is presented in Figure 1(a), and by strategy in Figure 1(b). We computed precision by considering all queries, also the ones having less than k results. There is not much difference between the obtained precision results at top-1 to top-5, all of them being around 0.9, which we consider to be a good precision for our lightweight integration approach. Most of the strategies have high precision, being notable that strategies $\mathcal{SUB}^1$ and $\mathcal{SUB}^5$ give the best results in our settings. Strategy $\mathcal{SUB}^3$ shows the worst precision results at top-1, which is an indicator that the usage of this strategy for our presented settings needs to be revised.



(a) Mean Precision at Top-K (b) Precision for Top-K

Fig. 1. Overall Precision

4 Related Work

Ranking is mostly known from Information Retrieval approaches which rank the results from structured queries, nevertheless, none to our knowledge combine rewriting queries using wildcards and partial mappings, with ranking. Ranking is computed on relaxed or malleable queries in [5], by considering the quantification

of the correlations existing between attributes (of duplicate entities detected in the data), and ranking higher the results based on relaxations using higher correlated attributes. This is similar to our idea to use the confidence of the mapping in computing the ranking function, but we don't require data access to detect correlations. In [6], a technique for ranking query results on the semantic web takes into consideration the inferencing processes that led to each result, where the relevance of the returned results is computed based upon the specificity of the links used when extracting information from the knowledge base. This approach is complementary to our approach, since the confidence values from the inferencing process could be an additional confidence to the computation of our ranking. [2] aims also at integrating distributed sources containing RDF data described by ontologies. An important difference to this approach is that only rewriting without relaxation of queries is produced, so, the ranking for the results of queries only considers the confidence of the used mappings. Due to space constraints many other related approaches had to be left out of this section.

5 Conclusions and Future Work

We presented an approach for ranking results of rewritten and relaxed queries executed over different repositories. We use the confidence of employed mappings in combination with heuristics which consider the amount and position of wildcards introduced. These factors are combined in a weighted fashion, to produce a ranking value for the results of executing the modified query on a specific information source. Our evaluations performed over real world datasets show the efficiency of our introduced ranking function.

The usage of the query ranking value for deciding on executing a query (or not) on a given information source is an interesting idea to be explored. It could serve for finding a trade-of between result precision and recall of unknown but relevant information. A future idea would be to take into account in the ranking computation a factor reflecting the confidence of the used data sources.

References

1. Stecher, R., Niederée, C., Nejdl, W.: Wildcards for lightweight information integration in virtual desktops. In: CIKM (2008)
2. Straccia, U., Troncy, R.: Towards distributed information retrieval in the semantic web: Query reformulation using the oMAP framework. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 378–392. Springer, Heidelberg (2006)
3. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: International World Wide Web Conference, New York, NY, USA (2007)
4. van Elst, L., Kiesel, M.: Generating and Integrating Evidence for Ontology Mappings. In: Motta, E., Shadbolt, N.R., Stutt, A., Gibbins, N. (eds.) EKAW 2004. LNCS (LNAI), vol. 3257, pp. 15–29. Springer, Heidelberg (2004)
5. Zhou, X., Gaugaz, J., Balke, W.T., Nejdl, W.: Query relaxation using malleable schemas. In: SIGMOD: International Conference on Management of Data (2007)
6. Stojanovic, N., Studer, R., Stojanovic, L.: An approach for the ranking of query results in the semantic web. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 500–516. Springer, Heidelberg (2003)