DivQ: Diversification for Keyword Search over Structured Databases

Elena Demidova[1], Peter Fankhauser[1, 2], Xuan Zhou[3] and Wolfgang Nejdl[1]
[1]L3S Research Center, Hannover, Germany
[2]Fraunhofer IPSI, Darmstadt Germany
[3]CSIRO ICT Centre, Australia
{demidova, fankhauser, nejdl}@L3S.de xuan.zhou@CSIRO.au

ABSTRACT
Keyword queries over structured databases are notoriously ambiguous. No single interpretation of a keyword query can satisfy all users, and multiple interpretations may yield overlapping results. This paper proposes a scheme to balance the relevance and novelty of keyword search results over structured databases. Firstly, we present a probabilistic model which effectively ranks the possible interpretations of a keyword query over structured data. Then, we introduce a scheme to diversify the search results by re-ranking query interpretations, taking into account redundancy of query results. Finally, we propose α-nDCG-W and WS-recall, an adaptation of α-nDCG and S-recall metrics, taking into account graded relevance of subtopics. Our evaluation on two real-world datasets demonstrates that search results obtained using the proposed diversification algorithms better characterize possible answers available in the database than the results of the initial relevance ranking.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Retrieval Models*

General Terms: Algorithms.

Keywords: diversity, ranking in databases, query intent.

1. INTRODUCTION
Diversification aims at minimizing the risk of user's dissatisfaction by balancing relevance and novelty of search results. Whereas diversification of search results on unstructured documents is a well-studied problem, diversification of search results over structured databases attracted much less attention. Keyword queries over structured data are notoriously ambiguous offering an interesting target for diversification. No single interpretation of a keyword query can satisfy all users, and multiple interpretations may yield overlapping results. The key challenge here is to give users a quick glance of the major plausible interpretations of a keyword query in the underlying database, to enable user to effectively select the intended interpretation.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

For example, a user who issued a keyword query "London" may be interested either in the capital of the United Kingdom or a book written by Jack London, an American author. In contrast to document search, where data instances need to be retrieved and analyzed, rich database structures offer a more direct and intuitive way of diversification. For instance, if keyword "London" occurs in two database attributes, such as "location" and "name", each of these occurrences can be viewed as a keyword interpretation with different semantics offering complementary results. In addition, as in a database the query disambiguation can be performed before the actual execution, the computational overhead for retrieving and filtering redundant search results can be avoided. In the final step the database system executes only the top-ranked query interpretations to retrieve relevant and diverse results.

Applying diversification techniques for unstructured documents to keyword queries over structured databases, calls for two main adaptations. First, keyword queries need to be interpreted in terms of the underlying database, such that the most likely interpretations are ranked on top. Second, diversification should take advantage of the structure of the database to deliver more diverse and orthogonal representations of query results. In this paper we present *DivQ* - a novel approach to search result diversification in structured databases. We first present a probabilistic query disambiguation model to create semantic interpretations of a keyword query over a structured database. Then, we propose a diversification scheme for generating the top-k most relevant and diverse query interpretations.

For evaluation, we propose α-nDCG-W and WS-recall, an adaptation of α-nDCG [6] and S-recall [4], to measure both relevance and novelty of search results. These two new metrics take into account graded relevance of subtopics, which is important for evaluating search results on structured data. We performed a user study to assess the quality of the disambiguation model and the diversification scheme. Our evaluation results on two real world datasets demonstrates that search results obtained using the proposed algorithms are able to better characterize possible answers available in the database than the results obtained by the initial relevance ranking.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 presents the diversification scheme. Section 4 introduces the new α-nDCG-W and WS-recall measures. Section 5 contains the results of our empirical investigation. Section 6 provides a conclusion.

2. RELATED WORK
Recently, a number of schemes have been proposed for diversifying results of document retrieval. Several evaluation

schemes for result diversification have also been introduced [1, 2, 4, 6, 9, 20]. Most of the techniques (e.g. [2]) perform diversification as a post-processing or re-ranking step of document retrieval. These techniques first retrieve relevant results and then filter or re-order the result list to achieve diversification. However, this approach can hardly be applied to structured databases, where retrieval of all relevant data is usually computationally expensive [11], especially when search results have to be obtained by joining multiple tables. In contrast, *DivQ* embeds diversification in the phase of query disambiguation, before retrieving any search results. This offers two advantages. Firstly, as the query interpretations generated during query disambiguation have clear semantics, they offer quality information for diversification. Secondly, we avoid the overhead of generation of all relevant results. Only the results of the top ranked interpretations are retrieved from the database.

Another conventional approach to achieve diversification is clustering or classification of search results. Both techniques group search results based on similarity, so that users can navigate to the right groups to retrieve the desired information. Clustering and classification have been applied to document retrieval [1, 15], image retrieval [18], and database query results [5, 14]. Similar to result re-ranking, clustering is usually performed as a post-processing step, and is computationally expensive. Moreover, it lacks semantic interpretations, making results less understandable by end users [10]. Although classification is more understandable,classes are usually pre-defined, without taking into account the intent of the actual query. The query interpretations of *DivQ* can be regarded as a special kind of classes which have well-defined semantics. In contrast to pre-defined classes, query interpretations are generated on the fly based on users' keyword queries. Thus, query interpretations are both query aware and understandable for end users. Most importantly, in contrast to existing work, we consider the similarity between query interpretations to avoid redundant search results. This enables us to further improve user satisfaction.

Chen et al. [4] employ pseudo-relevance feedback to achieve diversification of search results. Different from *DivQ*, they consider the intent of a query only tacitly. Wang et al. [20] focus on a theoretical development of the portfolio theory for document ranking. They propose to select top-n documents and order them by balancing the overall relevance of the list against its risk (variance). We believe that the portfolio technique can be adopted to compute diverse query interpretations in *DivQ* too.

Apart from diversification for document retrieval, little work has focused on diversification of search results over structured data. In [5] the authors propose to navigate SQL results through categorization, which takes into account user preferences. In [19], the authors introduce a pre-indexing approach for efficient diversification of query results on relational databases. As the categorization and diversification in both approaches is performed on the result level, these approaches are complementary to *DivQ* which conducts diversification on the interpretations of keyword queries without retrieving any search results.

Recent approaches to database keyword search [8, 13, 16, 17, 21] translate a keyword query into a ranked list of structured queries, also known as query interpretations, such that the user can select the one that represents her informational need. This disambiguation step is also a crucial step of *DivQ*. We utilize a similar probabilistic model as [8] for query disambiguation. However, the existing query disambiguation approaches consider only the likelihood of different query interpretations rather than their diversity. As a result, users with uncommon informational needs may not receive adequate results [4]. For example, if the majority of users who issued the keyword query "London" were interested in the guide of a city, the results referring to books written by Jack London may receive a low rank and even remain invisible to users. *DivQ* alleviates this problem by providing not only relevant but diverse query interpretations.

3. The Diversification Scheme

The user interface of our database keyword search is similar to that of faceted search. *DivQ* translates a keyword query to a set of structured queries, also known as query interpretations. Given a keyword query, a database can offer a broad range of query interpretations with various semantics. The ranked query interpretations work as facets and provide a quick overview over the available classes of results. These facets enable users to easily navigate over the result set, choose the query interpretations that are relevant to the specific informational need and click on them to retrieve the actual search results (like in [8]). To minimize the risk of user's dissatisfaction in this environment, diversification is required to provide a better overview of the probable query interpretations, rather than a ranking based only on relevance.

Table 1. Structured Interpretations for a Keyword Query

Keyword query: CONSIDERATION CHRISTOPHER GUEST			
Relevance	Top-3 interpretations ranking	Relevance	Top-3 interpretations diversification
0.9	A director CHRISTOPHER GUEST of a movie CONSIDERATION	0.9	A director CHRISTOPHER GUEST of a movie CONSIDERATION
0.5	A director CHRISTOPHER GUEST	0.4	An actor CHRISTOPHER GUEST
0.8	An actor CHRISTOPHER GUEST in a movie CONSIDERATION	0.2	A plot containing CHRISTOPHER GUEST of a movie
…	…	…	…

Table 1 gives an example of the query interpretations for the keyword query "CONSIDERATION CHRISTOPHER GUEST", once ranked only by relevance, and once re-ranked by diversification. In this scenario, both rankings provide several possible interpretations of the query, so that the user can choose the intended one. However, ranking by estimated relevance bears the danger of redundant results. For example, the results of the partial interpretation "A director CHRISTOPHER GUEST" which is ranked second in the top-3 ranking clearly overlap with the results of the complete query interpretation ranked first. In contrast, the diversified ranking shows a set of possible complementary interpretations with increased novelty of results.

3.1 Bringing Keywords into Structure

In the context of a relational database, a structured query is an expression of relational algebra. To translate a keyword query K to a structured query Q, *DivQ* first obtains a set of ***keyword interpretations*** $A_i{:}k_i$, which map each keyword k_i of K to an element A_i of an algebraic expression. *DivQ* then joins the keyword interpretations using a predefined ***query template*** T [3, 11], which is a structural pattern that is frequently used to query the database. We call the structured query resulting from the translation process described above a ***query interpretation***.

For instance, "CONSIDERATION CHRISTOPHER GUEST" is first translated into a set of keyword interpretations, which are "director:CHRISTOPHER", "director:GUEST" and "movie: CONSIDERATION". Then, these keyword interpretations are connected to a template "A director X of a movie Y" to form a query interpretation "A director CHRISTOPHER GUEST of a movie CONSIDERATION".

A ***query interpretation iscomplete*** if it contains interpretations for all keywords from the initial user query. Otherwise we talk about ***partial query interpretation***. Given a keyword query K, the ***interpretation space*** of K is the entire set of possible interpretations of K. In this paper, we focus on interpretations that retrieve non-empty results from the database.

3.2 Estimating Query Relevance

We estimate relevance of a query interpretation Q to the informational need of the user as the conditional probability $P(Q|K)$ that, given keyword query K, Q is the user intended interpretation of K. A query interpretation Q is composed of a query template T and a set of keyword interpretations $I=\{A_j{:}\{k_{j1},k_{jn}\} \mid A_j \in T,\ \{k_j,k_{jn}\} \subset K,\ \{k_{i1},k_{im}\} \cap \{k_{j1},k_{jn}\} = \{\}$ for $i \neq j\}$. Note that $\{k_{i1},k_{im}\}$ need not to be consecutive in the query. Thus, the probability $P(Q|K)$ can be expressed as:

$$P(Q \mid K) = P(I, T \mid K). \qquad (1)$$

To simplify the computation, we assume that (*i*) each keyword has one particular interpretation intended by the user; and (*ii*) the probability of a keyword interpretation is independent from the part of the query interpretation the keyword is not interpreted to. Based on these assumptions and Bayes' rule, we can transform Formula 1 to:

$$P(Q \mid K) \propto \left(\prod_{A_j \in T} P\left(A_j : \{k_{j1}, k_{jn}\} \mid A_j\right) \right) \times \left(\prod_{k_u \in K \cap k_u \notin Q} P_u \right) \times P(T), \qquad (2)$$

where $P(T)$ is the prior probability that the template T is used to form a query interpretation. $P(A_j{:}\{k_{j1},k_{jn}\}|A_j)$ represents the probability that, given that A_j is a part of a query interpretation, keyword interpretations $A_j{:}\{k_{j1},k_{jn}\}$ are also a part of the query interpretation. In case a keyword $k_u \in K$ is not mapped to any keyword interpretation in Q, we introduce a smoothing factor P_u, which is the probability that the user's interpretation of keyword k_u does not match any available attribute in the database.

$P(A_j{:}\{k_{j1},k_{jn}\}|A_j)$) can be estimated using attribute specific term frequency, i.e., the average number of occurrences of the keyword combination $\{k_1,k_{jn}\}$ in the attribute A_j. Note that, when $\{k_{j1},k_{jn}\}$ co-occur in an attribute A_j, the joint probability $P(A_j{:}\{k_{j1},k_{jn}\}|A_j)$ will usually be larger than the product of the marginal probabilities $P(A_j{:}\{k_{j1}\}|A_j)\ldots\ P(A_j{:}\{k_{jn}\}|A_j)$. Thus, query interpretations that bind more than one keyword to the same attribute, for example, a first name and a last name of a person to attribute "name", will get higher ranked than query interpretations that bind keywords to different attributes. P_u is a constant, whose value is smaller than the minimum probability of any existing keyword interpretation, such that the function assigns higher probabilities to complete query interpretations than to partial interpretations. $P(T)$ can be estimated as a frequency of the template's occurrence in the database query log. When the query log is not available, we assume all templates to be equally probable. As indicated in Section 3.1, query interpretations with an empty result are assigned zero probability. In this case the independence assumption (ii) used in Equation 2 is obviously violated, because the query interpretation maps keywords k_1 and k_2 to attributes A_1 and A_2, such that the marginal probabilities $P(A_1{:}\{k_1\}|A_1)$ and $P(A_2{:}\{k_2\}|A_2)$ are larger than zero, but, given the instances of the database, the joint probability $P(A_1{:}\{k_1\},A_2{:}\{k_2\}|A_1,A_2)$ is zero.

3.3 Estimating Query Similarity

As our objective is to obtain diverse query results, we want the resulting query interpretations to be not only relevant but also as dissimilar to each other as possible. Let Q_1 and Q_2 be two query interpretations of a keyword query K. Let I_1 and I_2 be the sets of keyword interpretations contained by Q_1 and Q_2 respectively. To assess similarity between the two query interpretations, we compute the Jaccard coefficient between I_1 and I_2.

Def. 1 (Query Similarity): We define similarity between two query interpretations Q_i, and Q_j as the Jaccard coefficient between the sets of keyword interpretations they contain, that is,

$$Sim(Q_1, Q_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|}. \qquad (3)$$

The resulting similarity value should always fall in [0, 1], where 1 stands for the highest possible similarity.

3.4 Combining Relevance and Similarity

To generate the top-k query interpretations that are both relevant and diverse, we employ a greedy procedure. We always select the most relevant interpretation as the first interpretation presented to the user. Then, each of the following interpretations is selected based on both its relevance and novelty. Namely, given a query interpretation Q and a set of query interpretations QI that are already presented to the user, we estimate the score of Q as its relevance score discounted by the average similarity between Q and all the interpretations in QI:

$$Score(Q) = \lambda \cdot P(Q \mid K) - (1-\lambda) \cdot \sum_{q \in QI} \frac{Sim(Q,q)}{|QI|}. \qquad (4)$$

Relevance and similarity factors are normalized to equal means before λ-weighting is applied. The interpretation with the highest score is selected as the next interpretation to be presented to the user. In Formula 4, λ is a parameter to trade-off query interpretation relevance against novelty. For example, with λ=1 the score of the query interpretation takes only relevance into account; λ= 0.5 corresponds to a balance between relevance and novelty, whereas λ<0.5 emphasizes novelty of the interpretation.

3.5 The Diversification Algorithm

To create a set R of the most relevant and diverse query interpretations in an efficient way we first materialize the top-k most probable query interpretations of a keyword query and sort the interpretations according to the relevance scores. Then we go

through the query interpretations and output the most relevant and diverse interpretations one by one. The pseudo-code of the algorithm is presented in Algorithm 1. Let L be the list of top-k query interpretations sorted by probability of their relevance to the user's informational need. The process starts with the most relevant query interpretation at the top of L. To compute the i^{th} relevant and diverse element, i.e. $R[i]$, we scan the remaining candidate elements in L, compare their scores in Formula 4, and add the element with the highest score to R. As the diversity value of each item is always larger than 0, it is not necessary to scan the entire L to obtain each $R[i]$. The scan stops when we are sure that the rest of L cannot possibly outperform the current optimal item, which is evaluated by "best_score > $\lambda P(L[j])$". The algorithm terminates after r elements are selected.

```
Input: list L[l] of top-k query interpretations ranked by relevance
Output: list R[r] of the relevant and diverse query interpretations
Proc Select Diverse Query Interpretations
R[0]=L[0]; i=1;
//less than r elements selected
while (i<r){
//select the best candidate for R[i]
        j=i; best_score=0;
        //more candidates for R[i]in L
        while(L[j]!=null){
                //check score upper bound
                if (best_score>λP(L[j])) break;
                if (score(L[j])>best_score){
                        best_score=score(L[j]);
                        c=j);
                } j++;
        }
        //add the best candidate to R
        R[i] =L[c];
        Swap L[i…c-1] and L[c];
        i++;
}
End Proc;
```

Algorithm 1. Select Diverse Query Interpretations

The worst case complexity of the Algorithm 1 is $O(l*r)$, where l is the number of query interpretations in L and r is the number of query interpretations in the result list R. The maximal total number of similarity computations is $(l^2-l)/2$.

4. EVALUATION METRICS

α-NDCG [6] and S-recall [4] are established evaluation metrics for document retrieval in presence of diversity and subtopics. As results of keyword search over structured data differ from conventional documents, these metrics require some adaptation.

A search result of *DivQ* is a ranked list of query interpretations. Therefore, a "document" in traditional IR corresponds to the union of tuples returned for one particular query interpretation in *DivQ*. Each tuple can be represented by its primary key in the database. Thus, a primary key corresponds to the notion of information nugget in α-NDCG and to subtopic in S-recall. However, the correspondence is loose: whereas α-NDCG and S-recall assume equal relevance of information nuggets and subtopics contained in a document, relevance of primary keys in a query result may vary a lot. Thus it is important to take into account their relevance for estimating gain and recall explicitly. In the following, we adapt α-NDCG and S-recall to this end.

4.1 Adapting Gain for α-NDCG-W

nDCG (normalized Discounted Cumulative Gain) has established itself as the standard evaluation measure when graded relevance values are available [12, 6]. The first step in the nDCG computation is creation of a gain vector G. The gain $G[k]$ at rank k can be computed as the relevance of the result at this rank to the user's keyword query. The gain may be discounted with increasing rank, to penalize documents lower in the ranking, reflecting the additional user effort required to reach them. The discounted gain is accumulated over k to obtain the DCG (Discounted Cumulative Gain) value and normalized using the ideal gain at rank k to finally obtain the nDCG value.

To balance relevance of search results with their diversity, the authors of [6] proposed α-nDCG, where the computation of the gain $G[k]$ is extended with a parameter α, representing a tradeoff between relevance and novelty of a search result. To assess novelty of a document in the search result, α-nDCG views a document as the set of information nuggets. If a document at rank i contains an information nugget n, α-NDCG counts how many documents containing n were seen before and discounts the gain of this document accordingly. α has a value in the interval [0, 1]; α=0 means that α-NDCG is equivalent to the standard nDCG measure. With increasing α, novelty is rewarded with more credit. When α is close to 1, repeated results are regarded as completely redundant such that they do not offer any gain. In [20], the authors fix α as 0.5 for a balance between relevance and novelty.

In the context of database keyword search, where an information nugget corresponds to a primary key, the relevance of nuggets with respect to the user query can vary a lot. To reflect the graded relevance assessment on the nuggets in the evaluation metrics, α-NDCG-W measures the gain G[k] of a search result at rank k as the relevance of the query interpretation at rank k, i.e., Q_k. We penalize the gain of an interpretation retrieving overlapping results using the following formula:

$$G[k] = relevance(Q_k) \cdot (1-\alpha)^r, \qquad (5)$$

where r is the factor, which expresses overlap in the results of the query interpretation Q_k with results of the query interpretations at ranks 1…k-1.

To compute r, for each primary key pk_i in the result of Q_k we count how many query interpretations with pk_i were seen before (i.e. at ranks 1…k-1), and aggregate the counts:

$$r = \sum_{pk_i \in Q_k} \sum_{j \in [1,k-1]} \left| pk_i \in Q_j \right|. \qquad (6)$$

Note that we consider primary keys in the result of one interpretation to be distinct (each primary key counts only once).

As in document retrieval the presence of a particular information nugget in a document is uncertain, the gain computation in [6] focuses on the number of nuggets contained in a document and does not take into account graded relevance of information nuggets. In contrast, in the context of database keyword search an information nugget in α-nDCG-W corresponds to a primary key in the result of a query interpretation, such that the presence of an information nugget in the result is certain. At the same time, relevance of retrieved primary keys with respect to the user query can vary a lot. This graded relevance is captured by Equation 5.

Agrawal et al. [1] suggest an alternative approach called NDCG-IA (for Intent Aware NDCG) to take into account graded relevance of information nuggets to queries. However, a drawback of NDCG-IA is that it may not lie between [0, 1]. Moreover, NDCG-IA does not take into account result overlap. In contrast, the value of relevance-aware α-nDCG takes into account result overlap and is always in the interval [0, 1], where 1 corresponds to ranking according to the user assessment of query interpretation relevance averaged over users.

4.2 Weighted S-Recall

Instance recall at rank k (S-recall) is an established recall measure which is applied when search results are related to several subtopics. S-recall is the number of unique subtopics covered by the first k results, divided by the total number of subtopics [4, 20].

In database keyword search, a single primary key in the search result corresponds to a subtopic in S-recall. However, other than in document retrieval, where all subtopics can be considered equally important, relevance of retrieved primary keys (tuples) can vary a lot with respect to the user query. To take the graded relevance of subtopics into account, we developed a WS-recall measure (weighted S-recall). WS-Recall is computed as the aggregated relevance of the subtopics covered by the top-k results (in our case query interpretations) divided by the maximum possible aggregated relevance when all relevant subtopics are covered:

$$WS-recall@k = \frac{\sum_{pk \in Q_{1..k}} relevance(pk)}{\sum_{pk \in U} relevance(pk)}, \qquad (7)$$

where U is the set of relevant subtopics (primary keys). In case only binary relevance assessments are available, WS-recall corresponds to S-recall. We average WS-recall at k and α-NDCG at k over a number of topics to get their means over the query set.

5. EXPERIMENTS

To assess the quality of the disambiguation and diversification schemes we performed a user study and a set of experiments.

5.1 Dataset and Queries

In our experiments, we used two real-world datasets: a crawl of the Internet Movie Database (IMDB) and a crawl of a lyrics database from the web. The IMDB dataset contains seven tables, such as movies, actors and directors, with more than 10,000,000 records. The Lyrics dataset contains five tables, such as artists, albums and songs, with around 400,000 records. As these datasets do not have any associated query log, we extracted the keyword queries from the query logs of MSN and AOL Web search engines. We pruned the queries based on their target URLs, and obtained thousands of queries for the IMDB and lyrics domains.

To obtain the most popular keyword queries, we first sorted the queries based on frequency of their usage in the log. For each domain, we selected 200 most frequent queries for which multiple interpretations with non-empty results exist in the database. These queries were mostly either single keyword or single concept queries, often referring to actor/artist names or movie/song titles. We refer to this part of the query set as *single concept queries (sc)*. To obtain an additional set of more complex queries, we manually selected about 100 queries for each dataset from the query log, where we explicitly looked for queries containing more than one concept, e.g. a movie/song title and an actor/artist name. We refer to this set as *multi-concept queries (mc)*.

As diversification of results is potentially useful for ambiguous queries [7], we estimated ambiguity of the resulting keyword queries using an entropy-based measure. To this end, for each keyword query, we ranked interpretations of this query available in the database using Formula 2 and computed the entropy in the top-10 ranks of the resulting list. Intuitively, given a keyword query, high entropy over the top ranked interpretations indicates potential ambiguity. Finally, we selected 25 single concept and 25 multi-concept queries with the highest entropy for each dataset.

5.2 User Study

To assess relevance of possible query interpretations we performed a user study. We selected a mix of single and multi-concept queries as described in Section 5.1, for which we generated all possible interpretations sorted by their probability. As the set of possible interpretations grows exponentially with the number of concepts involved, we took at most the top-25 interpretations. This should not rule out meaningful interpretations, as probabilities fall very quickly with their rank: Figures 1a and 1b give the maximum and the average ratio of the probability of a query at rank i and the aggregated probabilities of queries at rank $j<i$: $PR_i = P(Q_i \mid K) / \sum_{j<i} P(Q_j \mid K)$.

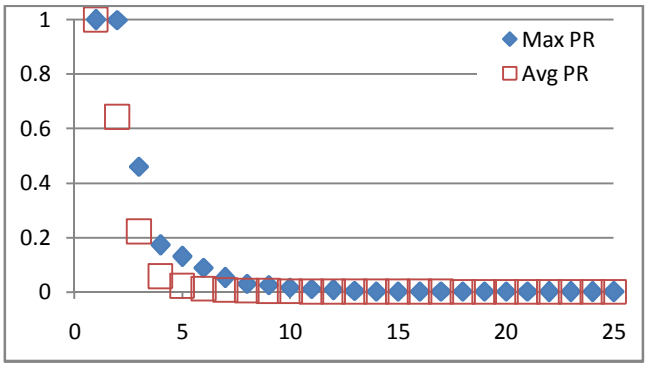

Figure 1a. Maximum and Average Probability Ratio, IMDB.

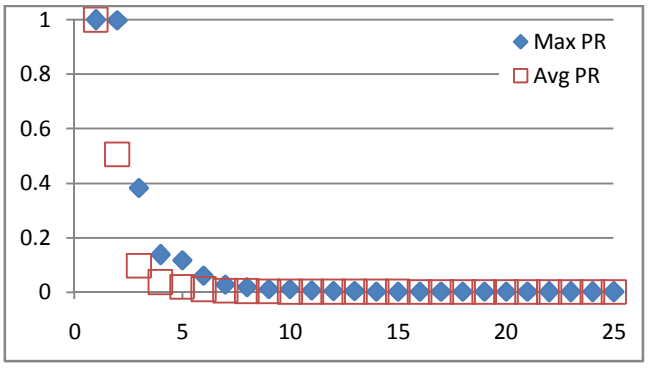

Figure 1b. Maximum and Average Probability Ratio, Lyrics.

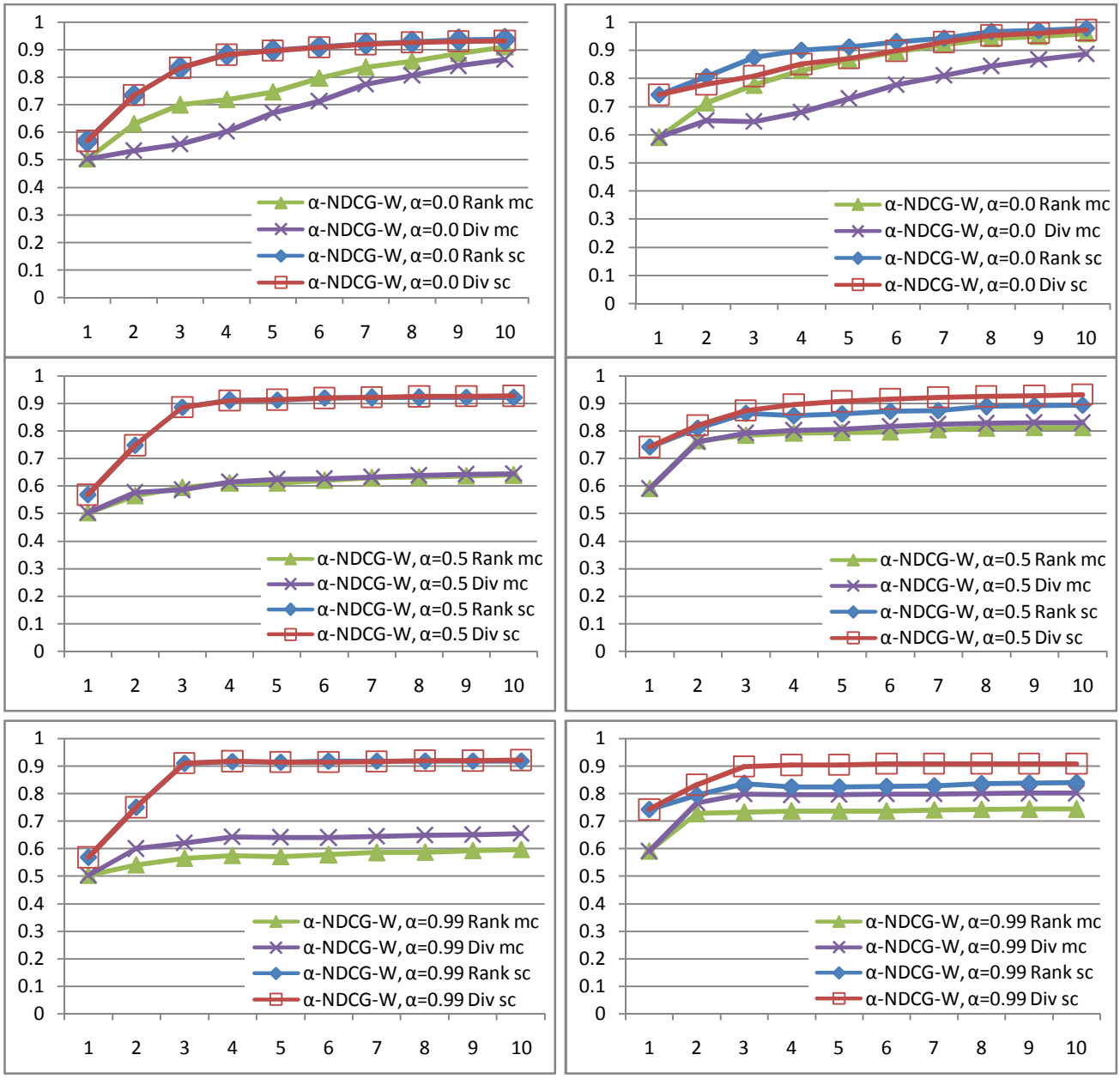

Figure 2a. α-NDCG-W, IMDB.

Figure 2b. α-NDCG-W, Lyrics.

(α-NDCG-W for single concept (sc) and multi-concept (mc) queries for diversification (Div) and ranking (Rank) for α=0, 0.5, and 0.99)

Each data point on the X-axis of Figures 1a and 1b presents the rank. The Y-axis presents the corresponding average and maximum PR_j value. As can be seen, queries at rank 10 already are only 0.01 as likely as queries at rank < 10, and queries at rank 25 are at most 2.95E-04 as likely as queries at rank < 25.

For each query we pruned all query interpretations Q_i whose probability constituted less than 0.1% of the aggregated probability of all possible interpretations. Additionally, for each query we included at most five more interpretations with probability below this threshold and randomized the order in which the interpretations were presented for user assessment, in order to avoid a bias towards top ranked queries.

In total, each user had to evaluate 630 interpretations for IMDB and 517 interpretations for the Lyrics dataset. For each interpretation of a given keyword query, the participants were asked to indicate on a two-point Likert scale, if they think that this interpretation could reflect an informational need implied by the keyword query. Multiple interpretations of one query were possible and explicitly encouraged. In total, we had 16 participants, from whom 10 completed all evaluation tasks in both datasets and the rest completed 30% of tasks in IMDB and 9% of tasks in lyrics dataset on average. We computed agreement between the participants using kappa statistics [15]. We observed average kappa values of 0.33 in IMDB and 0.28 in Lyrics. We consider this low agreement as an additional indication of

ambiguity of the selected queries. Finally, we computed the relevance scores of each query interpretation by averaging scores over the participants.

5.3 α-nDCG-W

Given a keyword query, *DivQ* first creates a ranked list of query interpretations and then applies the diversification algorithm to this list to obtain the most relevant and novel results. To assess quality of query ranking and diversification, we measured α-NDCG-W by varying α parameter from 0, to 0.5 and to 0.99. In the case of α=0, novelty of results is completely ignored, and α-NDCG-W corresponds to the standard NDCG. With α=0.5, novelty is given a certain credit. With α=0.99, novelty becomes crucial, and results without novelty are regarded as completely redundant. As the optimal ranking for normalization of DCG we ranked query interpretations by their user score. To achieve better overview of the available results in this experiment we set λ=0.1 (Equation 4); this enables the system to emphasize novelty of results in both datasets. We discuss the influence of λ value in Section 5.5. The results of the α-NDCG-W evaluation are presented in Figures 2a, 2b.

Each diagram of Figures 2a and 2b corresponds to a different α value. Each data point of the X-axis of a diagram represents the *k* for top-*k* query interpretations. The Y-axis represents the corresponding α-NDCG-W value. We use the symbol Rank to denote the ranking algorithm without diversification, and Div to denote the ranking algorithm with diversification. The α-NDCG-W values in the diagrams are averaged on single concept queries (*sc*) and multi-concept queries (*mc*) respectively. As we can see, in our experiments on the IMDB dataset (Figure 2a), the average α-NDCG-W for top-1 result of both ranking and diversification on single concept queries was always 0.58, given any value of α. For top-5 results, the gain of single concept queries increased to 0.9 in both datasets. For multi-concept queries, with α=0 the gain of ranking reaches 0.8 and 0.9 at top-6 in IMDB and Lyrics respectively. The relatively high α-NDCG-W values for α=0 confirm the quality of the ranking function.

As Figures 2a and 2b show, for α=0 ranking dominates diversification in all the cases. This is expected, as for α-values below 0.5, relevance is rewarded over novelty. In this case, diversification does not show its benefit. In the Lyrics dataset, the first benefits of diversification for single concept queries become visible already with α=0.5 at *k*=4, where α-NDCG-W improves by about 4%. This advantage increases with growing α, and achieves 8% at α=0.99. For single concept queries on IMDB, we did not observe any difference between ranking and diversification (the lines Rank sc and Div sc almost overlap in all diagrams of the Figure 2a). This is because the top query interpretations returned by ranking already deliver distinct results. In this case diversification preserves the high gain values achieved by ranking. For multi-concept queries, the gain of diversification grows with increasing α. When α=0.99 and k>3, diversification on mc queries outperforms ranking by about 7% in both datasets. The results of the paired ttest confirm statistical significance of this result for the confidence level of 95%. In summary, diversification performed on top of query ranking achieves significant reduction of result redundancy, while preserving retrieval quality in the majority of the cases.

5.4 WS-recall

We evaluate recall quality of the system using the WS-recall measure presented in Section 4.2. WS-recall computation requires user assessments of subtopic relevance. As graded relevance assessments of top query interpretations were available to us as a result of the user study, we compute relevance of a subtopic (primary key) as the relevance of the interpretation which returns this primary key. As one and the same primary key can be returned by multiple distinct query interpretations, we take the maximal score. As user judgments were available only for a subset of the interpretation space, the absolute recall values obtained by this approach might be too optimistic. However, they enable a fair comparison of the algorithms. We present the results of the WS-recall evaluation in Figures 3a and 3b.

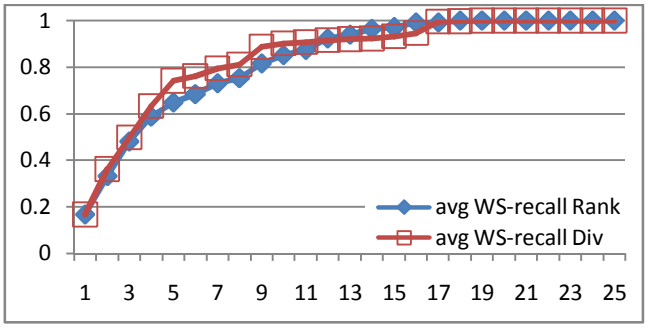

Figure 3a. WS-recall for Ranking and Diversification, IMDB.

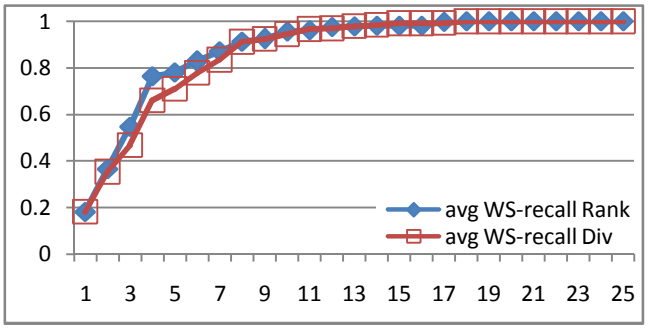

Figure 3b. WS-recall for Ranking and Diversification, Lyrics.

Each data point of the X-axis of Figures 3a and 3b corresponds to *k* for top-*k* interpretations. The Y-axis represents the corresponding WS-recall value of ranking (Rank) and diversification (Div) averaged over a set of queries. For example, in the Lyrics dataset (Figure 3b) the WS-recall of ranking increased from 0.2 in top-1 to 0.8 in top-6. As Figures 3a, 3b show, on average, ranking and diversification perform similar with respect to recall. We observed a slight improvement by diversification for *k*=2…11 in the IMDB dataset, whereas in Lyrics WS-recall at corresponding *k* values slightly decreased. Inspection of the actual query interpretations reveals that this is mainly due to the fact that ranking by relevance in Lyrics prefers complete query interpretations with large result sizes (i.e. large total number of returned tuples), whereas diversification pushes partial query interpretations with smaller result sizes. All other things equal, a larger result size increases WS-recall more. Normalizing result sizes for WS-recall is subject to future work.

In total we did not observe any significant effect of diversification on WS-recall values.

5.5 Balancing Relevance and Novelty

In Equation 4, λ is a parameter to balance query interpretation relevance against novelty. We evaluated influence of λ on α-NDCG-W at top-5 by α=0.99.

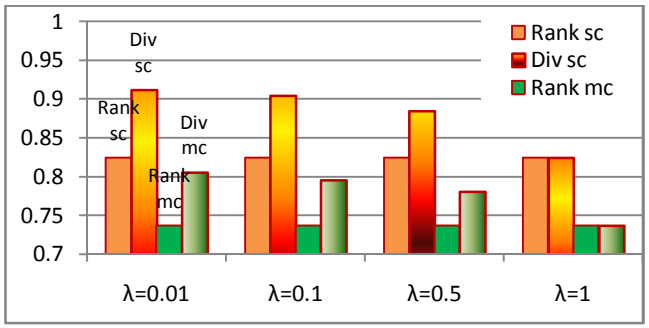

Figure 4. α-NDCG-W, k=5, α=0.99, Lyrics.

The results on the lyrics dataset are presented on Figure 4. The X-axis of Figure 4 presents the values of λ. The Y-axis represents the corresponding value of α-NDCG-W at top-5 by α=0.99. Each bar on Figure 4 presents α-NDCG-W for ranking and diversification of single concept (sc) and multi-concept (mc) queries averaged over a set of queries. For example, the average α-NDCG-W of diversification for single concept queries increased from 0.82 by λ=1 to 0.91 by λ=0.01. As can be seen, high α-NDCG-W values achieved by diversification of both, single concept and multi-concept queries decrease with increasing λ, until they meet α-NDCG-W of the original ranking in λ=1. The smaller the value of λ, the more visible is the impact of diversification and the more α-NDCG-W values of diversification outperform the original ranking. In contrast, with increasing λ, relevance of query interpretations dominates over novelty and the amount of re-ranking achieved by diversification becomes smaller. For example, for λ= [0.01, 0.5] the Spearman's rank correlation coefficient between the ranks of the query interpretations in the initial ranking and their ranks after diversification ranges between [0.74, 0.82] for single concept queries and [0.4, 0.67] for multi-concept queries in Lyrics. In the IMDB dataset multi-concept queries perform similar with rank correlation of [0.36, 0.69]. As different interpretations of single concept IMDB queries already deliver distinct results, we did not observe any significant re-ranking by varying λ on this query set.

6. CONCLUSION

In this paper we presented an approach to search result diversification over structured data. We introduced a probabilistic query disambiguation model to create relevant query interpretations over the structured data and evaluated the quality of the model in a user study. Furthermore, we proposed query similarity measure and a greedy algorithm to efficiently obtain relevant and diverse query interpretations. We proposed an adaptation of the established evaluation metrics to measure quality of diversification in database keyword search. Our evaluation results demonstrate the quality of the proposed model and show that using our algorithms the novelty of keyword search results over structured data can be substantially improved. Search results obtained using the proposed algorithms are also better characterize possible answers available in the database than the results obtained by the initial relevance ranking.

7. ACKNOWLEDGMENTS

This work is partially supported by the FP7 EU Projects OKKAM (contract ICT-215032) and LivingKnowledge (contract 231126).

8. REFERENCES

[1] Agrawal, R., Gollapudi, S., Halverson, A., & Leong, S. Diversifying Search Results. *WSDM 2009.*

[2] Carbonell, J., & Goldstein, J. The use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. *In Proceedings of the SIGIR 1998.*

[3] Chakaravarthy, V. T., Gupta, H., Roy, P., & Mohania, M. Efficiently Linking Text Documents with Relevant Structured Information. *In Proceedings of the VLDB 2006.*

[4] Chen, H., & Karger, D. R. Less is More. Probabilistic Models for Retrieving Fewer Relevant Documents. *SIGIR'06*

[5] Chen, Z., & Li, T. Addressing Diverse User Preferences in SQL-Query-Result Navigation. *SIGMOD 2007*

[6] Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I. Novelty and Diversity in Information Retrieval Evaluation. *SIGIR 2008.*

[7] Clough, P., Sanderson, M., Abouammoh, M., Navarro, S., Paramita, M.: Multiple Approaches to Analysing Query Diversity. *In Proceedings of SIGIR2009.*

[8] Demidova, E., Zhou, X. & Nejdl, W. IQ^P: Incremental Query Construction, a Probabilistic Approach. *In ICDE 2010.*

[9] Gollapudi, S., Sharma, A. An Axiomatic Approach for Result Diversification. *In Proceedings of WWW 2009.*

[10] Hearst, M. A. Clustering versus Faceted Categories for Information Exploration. *Commun, ACM 49, April 2006.*

[11] Hristidis, V., Gravano, L., Papakonstantinou, Y. Efficient IR-Style Keyword Search over Relational Databases. *VLDB 03.*

[12] Järvelin, K., & Kekäläinen, J. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst., 2002.*

[13] Kandogan, E., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., & Zhu, H. Avatar Semantic Search: A Database Approach to Information retrieval. *SIGMOD 2006.*

[14] Liu, B., & Jagadish, H. V. Using Trees to Depict a Forest. *In Proceedings of the VLDB 2009.*

[15] Manning, C. D., Raghavan, P. and Schütze, H. Introduction to Information Retrieval, *Cambridge University Press. 2008.*

[16] Tata, S., & Lohman, G. M. SQAK: doing more with keywords. *In Proceedings of the SIGMOD 2008.*

[17] Tran, T., Cimiano, P., Rudolph, S., & Studer, R. Ontology-based Interpretation of Keywords for Semantic Search. *In Proceedings of the ISWC 2007.*

[18] vanLeuken, R., Pueyo, L., Olivares, X., & Zwol, R. Visual Diversification of Image Search Results. *WWW 2009.*

[19] Vee, E., Srivastava, U., & Shanmugasund, J. Efficient Computation of Diverse Query Results. *ICDE 2008.*

[20] Wang, J., & Zhu, J. Portfolio Theory of Information Retrieval. *In Proceedings of the SIGIR 2009.*

[21] Zhou, Q., Wang, C., Xiong, M., Wang, H., Yu, Y. SPARK: Adapting Keyword Query to Semantic Search. *ISWC 2007.*