Diversification for Multi-domain Result Sets

Alessandro Bozzon, Marco Brambilla, Piero Fraternali, and Marco Tagliasacchi

Politecnico di Milano, Piazza Leonardo Da Vinci, 32 - 20133 Milano, Italy
{bozzon,mbrambil,fraterna,tagliasa}@elet.polimi.it

Abstract. Multi-domain search answers to queries spanning multiple entities, like "Find a hotel in Milan close to a concert venue, a museum and a good restaurant", by producing ranked sets of entity combinations that maximize relevance, measured by a function expressing the user's preferences. Due to the combinatorial nature of results, good entity instances (e.g., five stars hotels) tend to appear repeatedly in top-ranked combinations. To improve the quality of the result set, it is important to balance relevance with diversity, which promotes different, yet almost equally relevant, entities in the top-k combinations. This paper explores two different notions of diversity for multi-domain result sets, compares experimentally alternative algorithms for the trade-off between relevance and diversity, and performs a user study for evaluating the utility of diversification in multi-domain queries.

1 Introduction

Multi-domain search tries to respond to queries that involve multiple correlated concepts, like "*Find an affordable house in a city with low criminality index, good schools and medical services*" . Multi-domain search has the potential of bridging the gap between general purpose search engines, which are able to retrieve instances of at most one entity at a time (e.g., cities, products), and vertical search applications in specific domains (e.g., trip planning, real estate), which can correlate only a fixed set of information sources. Formally, multi-domain queries can be represented as rank-join queries over a set of relations, representing the wrapped data sources [11][14]. Each item in the result set is a combination of objects that satisfy the join and selection conditions, and the result set is ranked according to a scoring function, which can be expressed as a combination of local relevance criteria formulated on objects or associations (e.g., price or rating for a hotel, distance between the conference venue, hotel, and restaurant). Due to the combinatorial nature of multi-domain search, the number of combinations in the result set is normally very high, and strongly relevant objects tend to combine repeatedly with many other concepts, requiring the user to scroll down the list of results deeply to see alternative, maybe only slightly less relevant, objects.

As a running example, consider a multi-domain search scenarios where three data sources are wrapped by the following relations: *Hotel*(`HName`, `HLoc`, `HRating`, `HPrice`), *Restaurant*(`RName`, `RLoc`, `RRating`, `RPrice`), *Museum*(`MName`, `MLoc`, `MRating`, `MPrice`).

M. Brambilla, T. Tokuda, and R. Tolksdorf (Eds.): ICWE 2012, LNCS 7387, pp. 137–152, 2012.

Table 1. Top-K result set based on relevance

	Hotel		*Restaurant*		*Museum*		$S^{(a)}(\tau, q)$
	HName	HPrice	RName	RPrice	MName	MPrice	Total price
τ_1	Hotel Amadeus	€35	Miyako	€25	Galleria d'Arte Moderna	€0	€60
τ_2	Hotel Amadeus	€35	Miyako	€25	Museo Civico di Milano	€0	€60
τ_3	Hotel Amadeus	€35	Miyako	€25	Museo di Storia Contemp.	€0	€60
τ_4	Hotel Amadeus	€35	Porca Vacca	€25	Galleria d'Arte Moderna	€0	€60
τ_5	Hotel Amadeus	€35	Porca Vacca	€25	Museo Civico di Milano	€0	€60
τ_6	Hotel Amadeus	€35	Porca Vacca	€25	Orto Botanico di Brera	€0	€60
τ_7	Hotel Amadeus	€35	Spontini 6	€25	Galleria d'Arte Moderna	€0	€60
τ_8	Hotel Amadeus	€35	Spontini 6	€25	Museo Civico di Milano	€0	€60
τ_9	Hotel Amadeus	€35	Spontini 6	€25	Orto Botanico di Brera	€0	€60
τ_{10}	Hotel Amadeus	€35	Spontini 6	€25	Museo di Storia Contemp.	€0	€60

A query issued with a mobile phone aims at finding combinations within a short distance from the user location and good ratings, to be returned in order of total price. If we suppose to find 100 hotels and restaurants and 20 events, and assume a 10% selectivity of the join and selection condition on distance and minimum rating, a total number of 20000 combinations can be used to build the top-K result set. Supposing to show only 10 combinations disregarding the relevance of the constituent objects, up to 30 distinct objects out of 220 can be presented (14%). However, in real situations, the composition of the top-K results also depends on relevance, which decreases diversity.

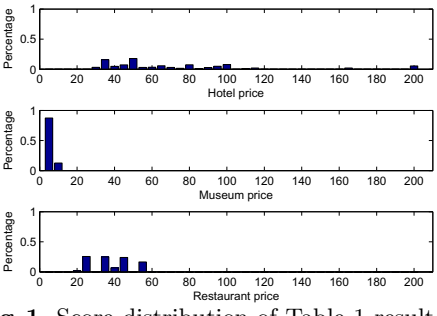

Fig. 1. Score distribution of Table 1 result set

For illustration, Table 1 shows a result set, which contains the top-10 combinations ranked according to total price. We observe that the result is rather poor in terms of diversity, as only 1 hotel, 3 restaurants and 4 museums are represented. Indeed, the number of distinct objects that appear in the top-K results is sensitive to the distribution of attribute values used to compute the score of the combination. In our case, the price range of good-rated hotels is larger than the price ranges of restaurants and events, as illustrated in Figure 1. Hence, budget hotels will appear repeatedly in the top-K list, lowering the number of distinct objects seen by the user. The same observation applies when, fixed an hotel, one considers the price range of restaurants compared to the price range of museums. This empirical observation is supported by Theorem 1 discussed later in this paper.

Improving the diversity of the result set is the aim of *diversification*, which can be defined in the context of multi-domain search as the selection of K elements out of a universe of N combinations, so to maximize a quality criterion that combines the *relevance* and the *diversity* of the objects of distinct types seen by the user. In this respect, Table 2 shows an example of result set with diversified combinations. We observe that the set does not necessarily contain the top-10

Table 2. Top-K result set based on relevance and diversity

	Hotel		*Restaurant*		*Museum*		$S^{(a)}(\tau, q)$
	HName	HPrice	RName	RPrice	MName	MPrice	Total price
τ_1	Hotel Amadeus	€35	Miyako	€25	Galleria d'Arte Moderna	€0	€60
τ_2	Hotel Amadeus	€35	Porca Vacca	€25	Museo di Storia Contemp.	€0	€60
τ_3	Hotel Amadeus	€35	Miyako	€25	Orto Botanico di Brera	€0	€60
τ_4	Hotel Nazioni	€36	Miyako	€25	Galleria d'Arte Moderna	€0	€61
τ_5	Hotel Nazioni	€36	The Dhaba	€25	Orto Botanico di Brera	€0	€61
τ_6	Hotel Nazioni	€36	Spontini 6	€25	Pad. d'Arte Contemp.	€2	€63
τ_7	Hotel Zefiro	€39	Matto Bacco	€25	Galleria d'Arte Moderna	€0	€64
τ_8	Hotel Zefiro	€39	Porca Vacca	€25	Museo Civico di Milano	€0	€64
τ_9	Hotel Nazioni	€36	Porca Vacca	€25	Museo della Perma.	€6	€67
τ_{10}	Hotel Zefiro	€39	Miyako	€25	Museo di Storia Naturale	€3	€67

combinations in terms of total price. Nevertheless, the result is much richer: 3 hotels, 5 restaurants and 7 museums are selected.

The contributions of the paper can be summarized as follows: (**a**) We discuss the problem of diversification in the context of multi-domain search, an area made quite interesting by the increasing availability of public "joinable" Web data sources. (**b**) We formalize multi-domain diversification and propose two criteria for comparing combinations (*categorical* and *quantitative* diversity). (**c**) Result diversification is NP-hard also in the multi-domain context; we therefore experimentally study the behavior of three known greedy algorithms, testing the hypothesis that the diversification algorithms improve the quality of the result set with respect to a baseline constituted by the selection of the most relevant K combinations. (**d**) We formally analyze under which conditions diversification can be potentially effective in improving the quality of the results. In particular, we consider the impact of the score distributions on the diversity of the result set which includes the top-K combinations based on relevance only. (**e**) We evaluate the perception and utility of diversification in multi-domain search with a user study that focuses on explicit comparison of result sets diversified according to different algorithms.

The organization of the paper is as follows: Section 2 presents the formalization of the problem and introduces different diversity measures for combinations; Section 3 illustrates the results of the experimental activity; Section 4 discusses previous work; Section 5 concludes and discusses future work.

2 Multi-domain Diversification

Consider a set of relations $R_1, R_2, \ldots, R_n$, where each R_i denotes the result set returned by invoking a search service σ_i over a Web data source. Each tuple $t_i \in R_i =< a_i^1, a_i^2, \ldots, a_i^{m_i} >$ has schema $R_i(\mathtt{A}_i^1 : \mathtt{D}_i^1, \ldots, \mathtt{A}_i^{m_i} : \mathtt{D}_i^{m_i})$, where $\mathtt{A}_i^{m_i}$ is an attribute of relation R_i and $\mathtt{D}_i^{m_i}$ is the associated domain. As usual in measurement theory, we distinguish the domains $\mathtt{D}_i^k$ into *categorical*, when values admit only equality test, and *quantitative*, when values can be organized in vectors embedded in a metric space.

A multi-domain query over the search services $\sigma_1, \ldots \sigma_n$ is defined as a join query $q = R_1 \bowtie \ldots \bowtie R_n$ over the relations $R_1, \ldots, R_n$, where the join

predicate can be arbitrary. We call *combination* an element of the join $\tau = t_1 \bowtie \cdots \bowtie t_n =< a_1^1, a_1^2, \ldots, a_1^{m_1}, \ldots, a_n^1, a_n^2, \ldots, a_n^{m_n} >$, and *result set* $\mathcal{R}$ the set of combinations satisfying the query q.

2.1 Relevance

The goal of multi-domain search is to support the user in selecting one or more combinations from the result set of a multi-domain query, so to maximize the satisfaction of his information seeking task. To this end, combinations can be presented in order of relevance. The relevance of a combination with respect to the query q can be expressed quantitatively by means of a user-defined (possibly non-monotone) *relevance score function* $S(\tau, q)$, which can be assumed to be normalized in the $[0, 1]$ range, where 1 indicates the highest relevance. When the result set $\mathcal{R}$ is sorted, e.g. in descending order of relevance, τ_k indicates the k-th combination of $\mathcal{R}$.

Example 1. With respect to the relations introduced in Section 1, consider a function *city()*, which returns the city where the geographical coordinates of a location belong to, and the multi-domain query $q =$ select * from *Hotel*, *Restaurant*, *Museum* where $city(\texttt{HLoc}) = \texttt{Milan} \wedge city(\texttt{RLoc}) = city(\texttt{HLoc}) \wedge city(\texttt{MLoc}) = city(\texttt{HLoc})$. The following relevance score functions could be used to rank hotel, restaurant and museum triples: (**a**) The overall price of the combination: $S^{(a)}(\tau, q) = sum(\texttt{HPrice}[t_h], \texttt{RPrice}[t_r], \texttt{MPrice}[t_m])$. (**b**) The average rating for the hotel and the restaurant, $S^{(b)}(\tau, q) = avg(\texttt{HRating}[t_h], \texttt{RRating}[t_r])$. (**c**) The distance of the shortest path that connects the hotel, the restaurant and the museum: $S^{(c)}(\tau, q) = f_{distance}(\texttt{HLoc}[t_h], \texttt{RLoc}[t_r], \texttt{MLoc}[t_m])$.

Note that $S^{(a)}$ and $S^{(b)}$ are simple linear (thus monotone) functions based solely on a subset of the attribute values of the constituent tuples, whereas $S^{(c)}$ uses a more complex function, not necessarily monotone, that might incorporate external knowledge (e.g. road maps).

2.2 Diversity

An implicit goal of multi-domain search is to present to the user a set of combinations that expresses a good coverage of the population of the constituent entities. Coverage can be improved by avoiding in the result set combinations that are too similar, according to some definition of similarity. Two different criteria can be employed to express the similarity (or symmetrically, the diversity) of combinations: (**a**) *Categorical diversity*: two combinations are compared based on the equality of the values of one or more categorical attributes of the tuples that constitute them. As a special case, categorical diversity can be based on the key attributes: this means evaluating if an object (or sub-combination of objects) is repeated in the two combinations. (**b**) *Quantitative diversity*: the diversity of two combinations is defined as their distance, expressed by some metric function.

In both cases, for each pair of combinations τ_u and τ_v, it is possible to define a diversity measure $\delta : \mathcal{R} \times \mathcal{R} \to [0, 1]$, normalized in the $[0, 1]$ interval, where 0 indicates maximum similarity.

Definition 1. *Let* $A_i^{j_{i,1}}, \ldots, A_i^{j_{i,d_i}}$ *be a subset of* d_i *attributes of relation* R_i *and* $\mathbf{v}_i(\tau) = [a_i^{j_{i,1}}, \ldots, a_i^{j_{i,d_i}}]^T$ *the projection of a combination* τ *on such attributes. Categorical diversity is defined as follows:*

$$\delta(\tau_u, \tau_v) = 1 - \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\mathbf{v}_i(\tau_u) = \mathbf{v}_i(\tau_v)} \tag{1}$$

where n *is the number of relations* R_i *and* $\mathbb{1}$ *is the indicator function, returning one when the predicate is satisfied.*

Intuitively, categorical diversity is the percentage of tuples in two combinations that do not coincide on the attributes $\mathtt{A}_i^{j_{i,1}}, \ldots, \mathtt{A}_i^{j_{i,d_i}}$. When these attributes are the key, categorical diversity can be interpreted as the percentage of objects that appear only in one of the two combinations.

Definition 2. *Let* $\mathbf{v}_i(\tau)$ *be as in Definition 1. Let* $\mathbf{v}(\tau) = [\mathbf{v}_1(\tau), \ldots, \mathbf{v}_n(\tau)]^T = [v_1(\tau), \ldots, v_d(\tau)]^T$ *denote the concatenation of length* $d = d_1 + \ldots + d_n$ *of such vectors. Given* d *user-defined weights* $w_1 \ldots, w_d$ *and a normalization constant* δ_{max}*, quantitative diversity is defined as:*

$$\delta(\tau_u, \tau_v) = \frac{1}{\delta_{max}} \sqrt[p]{\sum_{l=1}^{d} w_l |v_l(\tau_u) - v_l(\tau_v)|^p} \tag{2}$$

Quantitative diversity between combinations τ_u, τ_v is formalized as the (weighted) l_p-norm of the difference between the vectors $\mathbf{v}(\tau_u)$ and $\mathbf{v}(\tau_v)$. The normalization constant can be chosen, e.g., as the maximum distance value between pairs of combinations in the result set.

Example 2. A categorical diversity function can be computed by defining $\mathbf{v}_h(\tau) = [\mathtt{HName}(\tau)]$, $\mathbf{v}_r(\tau) = [\mathtt{RName}(\tau)]$ and $\mathbf{v}_m(\tau) = [\mathtt{MName}(\tau)]$. As an example, consider the combinations of Table 2. Then, $\delta(\tau_1, \tau_2) = 2/3$, $\delta(\tau_1, \tau_3) = 1/3$ and $\delta(\tau_1, \tau_6) = 1$.

A quantitative diversity function based on the hotel, restaurant and museum rating attributes can be defined as follows: let $p = 1$, $w_l = 1$, and $\mathbf{v}_1(\tau) = [\mathtt{HRating}(\tau)]$, $\mathbf{v}_2(\tau) = [\mathtt{RRating}(\tau)]$ and $\mathbf{v}_3(\tau) = [\mathtt{MRating}(\tau)]$. As an example, consider the combinations of Table 2. Then, $\mathbf{v}(\tau_1) = [35, 25, 0]$, $\mathbf{v}(\tau_3) = [36, 25, 0]$ and $\mathbf{v}(\tau_6) = [36, 25, 2]$. Then $\delta(\tau_1, \tau_3) = 1/\delta_{\max}$, $\delta(\tau_1, \tau_6) = 3/\delta_{\max}$ and $\delta(\tau_3, \tau_6) = 2/\delta_{\max}$.

2.3 Computing Relevant and Diverse Combinations

Based on the notion of diversity, it is possible to address the problem of extracting from the result set of a multi-domain query the top-K most relevant and

diverse combinations. Let $N = |\mathcal{R}|$ denote the number of combinations in the result set and $\mathcal{R}_K \subseteq \mathcal{R}$ the subset of combinations that are presented to the user, where $K = |\mathcal{R}_K|$. We are interested in identifying a subset $\mathcal{R}_K$ which is both *relevant* and *diverse*. Fixing the relevance score $S(\cdot, q)$, the dissimilarity function $\delta(\cdot, \cdot)$, and a given integer K, we aim at selecting a set $\mathcal{R}_K \subseteq \mathcal{R}$ of combinations, which is the solution of the following optimization problem [8]:

$$\mathcal{R}_K^* = \underset{\mathcal{R}_K \subseteq \mathcal{R}, |\mathcal{R}_K| = K}{\operatorname{argmax}} F(\mathcal{R}_K, S(\cdot, q), \delta(\cdot, \cdot)) \tag{3}$$

where $F(\cdot)$ is an objective function that takes into account both relevance and diversity. Two commonly used objective functions are `MaxSum` (Equation 4) and `MaxMin` (Equation 5), as defined in [8]

$$F(\mathcal{R}_K) = (K-1) \sum_{\tau \in \mathcal{R}_K} S(\tau, q) + 2\lambda \sum_{\tau_u, \tau_v \in \mathcal{R}_K} \delta(\tau_u, \tau_v) \tag{4}$$

$$F(\mathcal{R}_K) = \min_{\tau \in \mathcal{R}_K} S(\tau, q) + \lambda \min_{\tau_u, \tau_v \in \mathcal{R}_K} \delta(\tau_u, \tau_v) \tag{5}$$

where $\lambda > 0$ is a parameter specifying the trade-off between relevance and diversity.

Solving problem (3) when the objective function is given in (4) or (5) is NP-hard, as it can be reduced to the minimum k-center problem [9]. Nevertheless, greedy algorithms exist [8], which give a 2-approximation solution in polynomial time. Algorithm 1 and Algorithm 2 give, respectively, the greedy algorithms for `MaxSum` and `MaxMin`. In both cases the underlying idea is to create an auxiliary function $\delta'(\cdot, \cdot)$ and iteratively construct the solution by incrementally adding combinations in such a way as to locally maximize the given objective function.

Algorithm 1. Greedy algorithm for `MaxSum`.

Input : Set of combinations $\mathcal{R}$, K
Output: Selected combinations $\mathcal{R}_K$

begin
  Define $\delta'(\tau_u, \tau_v) = S(\tau_u, q) + S(\tau_v, q) + 2\lambda\delta(\tau_u, \tau_v)$
  Initialize the set $\mathcal{R}_K = \emptyset$, $U = \mathcal{R}$
  for $c = 1 : \lfloor K/2 \rfloor$ **do**
    Find $(\tau_u, \tau_v) = \underset{x,y \in U}{\operatorname{argmax}}\, \delta'(x, y)$ Set $\mathcal{R}_K = \mathcal{R}_K \cup \{\tau_u, \tau_v\}$ Set $U = U \setminus \{\tau_u, \tau_v\}$
  end
  If K is odd, add an arbitrary combination to $\mathcal{R}_K$
end

Another objective function, closely related to the aforementioned ones, is *Maximal Marginal Relevance* (`MMR`), initially proposed in [1]. Indeed, `MMR` implicitly maximizes an hybrid objective function whereby relevance scores are summed together, while the minimum distance between pairs of objects is controlled.

The algorithm originally proposed in [1] is identical to Algorithm 2, where line 6 is replaced by

$$\tau^* = \underset{\tau \in \mathcal{R} n \mathcal{R}_K}{\operatorname{argmax}} \left\{ S(\tau, q) + \lambda \min_{x \in \mathcal{R}_K} \delta(\tau, x) \right\} \tag{6}$$

and at line 4 the set $\mathcal{R}_K$ is initialized with the most relevant combination.

Algorithm 2. Greedy algorithm for `MaxMin`.

Input : Set of combinations $\mathcal{R}$, K
Output: Selected combinations $\mathcal{R}_K$

```
1 begin
2     Define δ'(τ_u, τ_v) = 1/2 (S(τ_u, q) + S(τ_v, q)) + λδ(τ_u, τ_v)
3     Initialize the set R_K = ∅
4     Find (τ_u, τ_v) = argmax_{x,y∈R} δ'(x, y) and set R_K = {τ_u, τ_v}
5     while |R_K| < K do
6         τ* = argmax_{τ∈R\R_K} min_{x∈R_K} δ'(τ, x)
7         Set R_K = R_K ∪ {τ*}
8     end
9 end
```

Note that for $\lambda = 0$ all algorithms return a result set which consists of the top-K combinations with the highest score, thus neglecting diversity.

2.4 When Diversification Helps

The score function and the diversity function both work on the attribute values of tuples. The question arises about the circumstances in which the ranking function alone would already guarantee a sufficiently varied result set, thus lowering the utility of diversification. The intuition is that when the attributes used in the ranking function have values distributed with comparable variance in the input relations, then the relevance score performs better at sampling the population than when attribute values are distributed with large variance differences. The following result formalizes this intuition and provides a guideline for deciding if diversification is needed. For simplicity, we consider two relations R_1 and R_2, with a population of N_1 and N_2 tuples, respectively.

Theorem 1. *Given a positive integer $K \in \mathbb{N}^+$, a score function $S(\tau, q) = w_1 s_1(t_1) + w_2 s_2(t_2)$, a set $\mathcal{R}_K$ that contains the K combinations with the highest score $S(\tau, q)$. Let $D_1(K)$ (resp. $D_2(K)$) be the expected number of distinct tuples of relation R_1 (resp. R_2) represented in $\mathcal{R}_K$. If the values of the score functions $s_1(), s_2()$ are uniformly distributed in intervals of width Δ_1 (resp. Δ_2), then $D_1(K)/D_2(K) = (w_2 \Delta_2 N_1)/(w_1 \Delta_1 N_2)$.*

Proof. The value of the local score $s_i = s_i(t_i)$ can be regarded as sampled from a probability density function $p_{s_i}(s_i)$. We want to determine the and $D_1(K)/D_2(K)$ given the score distributions $p_{s_1}(s_1)$ and $p_{s_2}(s_2)$.

Let $n_i(s_i)$ denote the number of tuples in R_i that exceed the value of s_i. Note that, given a deterministically chosen score value $\tilde{s}_i$, $n_i(\tilde{s}_i)$ is a discrete random variable, whose expected value can be expressed as

$$E[n_i(\tilde{s}_i)] = \bar{n}_i(\tilde{s}_i) = N_i\,(1 - P_{s_i}(\tilde{s}_i)) \tag{7}$$

where $P_{s_i}(\tilde{s_i})$ is the cumulative density function of the random variable s_i evaluated at $\tilde{s_i}$. The value of $D_i(\theta)$ (number of tuples of R_i contributing to the top combinations, i.e. those whose score $s_1 + s_2$ exceeds θ) can be determined as follows

$$\begin{aligned} D_i(\theta) &= \sqrt{\beta} E\left[\bar{n}_i(s_i) |\, s_1 + s_2 > \theta\right] \\ &= \sqrt{\beta} \int_{-\infty}^{+\infty} \bar{n}_i(s_i) p_{s_i}(s_i | s_1 + s_2 > \theta) ds_i \end{aligned} \tag{8}$$

where $\beta \in [0, 1]$ denotes the join selectivity, i.e. $|R_1 \bowtie R_2|/|R_1 \times R_2|$, assuming that the join predicate does not depend on the scores. In order to determine $p_{s_i}(s_i|s_1 + s_2 > \theta)$ we leverage Bayes's theorem

$$p_{s_i}(s_i|s_1 + s_2 > \theta) = \frac{p_{s_i}(s_1 + s_2 > \theta|s_i) p_{s_i}(s_i)}{\Pr\{s_1 + s_2 > \theta\}} \tag{9}$$

where

$$\begin{aligned} \Pr\{s_1 + s_2 > \theta\} &= p(s_2 > \theta - s_1) = 1 - P_{s_2}(\theta - s_1) \\ &= p(s_1 > \theta - s_2) = 1 - P_{s_1}(\theta - s_2) \end{aligned} \tag{10}$$

and

$$p(s_1 + s_2 > \theta) = 1 - P_{s_1+s_2}(\theta) \tag{11}$$

Therefore

$$D_i(\theta) = \sqrt{\beta} \int_{-\infty}^{+\infty} N_i\,(1 - P_{s_i}(s_i))\, \frac{(1 - P_{\bar{s}_i}(\theta - s_i))\, p_{s_i}(s_i)}{1 - P_{s_1+s_2}(\theta)} \tag{12}$$

Rewriting the previous expression, we obtain:

$$\begin{aligned} D_i(\theta) = \sqrt{\beta} \frac{N_i}{1 - P_{s_1+s_2}(\theta)} \int_{-\infty}^{+\infty} & p_{s_i}(s_i)\,(1 - P_{s_i}(s_i)) \cdot \\ & \cdot \left(1 - P_{s_{\bar{i}}}(\theta - s_i)\right) ds_i \end{aligned} \tag{13}$$

where $\bar{P}_{s_i}(s_i) = 1 - P_{s_i}(s_i)$ and the integral can be compactly written in terms of a convolution product, that is, $[(\bar{P}_{s_i} \cdot p_{s_i}) * \bar{P}_{s_{\bar{i}}}](\theta)$.

The value of $D_i(\theta)$ can be readily evaluated for simple distributions, e.g. uniform distributions of the scores, i.e. $s_i \sim U[\underline{s}_i, \bar{s}_i]$. In this case, $\theta \in [\theta_{\min}, \theta_{\max}] = [\underline{s}_1 + \underline{s}_2, \bar{s}_1 + \bar{s}_2]$. Note that, for each value of θ, the expected number of combinations whose score exceeds θ is given by

$$K(\theta) = \frac{\beta}{N_1 N_2}\,(1 - P_{s_1+s_2}(\theta)) \tag{14}$$

We are interested to the case when K is small, thus $\theta \simeq \theta_{\max}$. Let $\Delta_i = \bar{s}_i - \underline{s}_i$. When $\theta \in [\max\{\theta_{\max} - \Delta_1, \theta_{\max} - \Delta_2\}, \theta_{\max}]$, the integral in (13) evaluates to

$$D_i(\theta) = \frac{\sqrt{\beta} N_i}{1 - P_{s_1+s_2}(\theta)} \frac{1}{24 \Delta_i^2 \Delta_{\bar{i}}} \left(\bar{s}_1 + \bar{s}_2 + \theta\right)^3 \tag{15}$$

Therefore, the ratio $D_1(\theta)/D_2(\theta)$ simplyfies to

$$\frac{D_1(\theta)}{D_2(\theta)} = \frac{N_1}{N_2} \frac{\Delta_2}{\Delta_1} \tag{16}$$

Since there is a one-to-one mapping between K and θ,

$$\frac{D_1(K)}{D_2(K)} = \frac{D_1(\theta)}{D_2(\theta)} = \frac{N_1}{N_2} \frac{\Delta_2}{\Delta_1} \tag{17}$$

The case in which the scoring function is weighted, i.e. $w_1 s_1 + w_2 s_2$, can be reduced to the result above, by defining a new random variable $\hat{s}_i = w_i s_i$, such that $\hat{s}_i \sim U[w_i \underline{s}_i, w_i \bar{s}_i]$ and $\hat{\Delta}_i = w_i \Delta_i$. Therefore:

$$\frac{D_1(K)}{D_2(K)} = \frac{N_1}{N_2} \frac{w_2 \Delta_2}{w_1 \Delta_1} \tag{18}$$

According to Theorem 1 the ratio of the number of tuples from relation R_1 and R_2 in the top-K combinations is inversely proportional to the ratio of the score ranges. The result holds both when combinations are computed with the Cartesian product and with a join with arbitrary predicate. This observation can be used to determine the self-diversification power implicit in the distribution of data and in the ranking function and, indirectly, to assess the expected improvement that can be obtained with diversification, with respect to the case in which only the relevance score function is used to build the result set.

Say that $N_1 = N_2$ and $w_1 = w_2$; if if the variance of the scores is the same in both input relations (i.e. $\Delta_1 = \Delta_2$) then $D_1(K) = D_2(K)$, i.e. the number of distinct tuples extracted from R_1 and R_2 in the top combinations is the same, regardless the average score (i.e., even if $E[s_1] \neq E[s_2]$). Therefore, the result set computed based on relevance only already picks up tuples evenly from the input relations and, if the sizes of the populations are comparable, diversification is not expected to help much.

If, instead, the number of distinct tuples extracted from R_1 contributing to the top combinations is Δ_2/Δ_1 larger than the number of tuples extracted from R_2 (i.e. $\Delta_1 < \Delta_2$), then $D_1(K) > D_2(K)$, i.e. tuples coming from the distribution characterized by the largest variance tend to be under-represented in the top combinations. Hence, diversification is expected to help, especially when $\Delta_2 \gg \Delta_1$ or, viceversa, $\Delta_1 \gg \Delta_2$.

3 Experiments

Multi-domain search deals with *combinations* of objects; therefore, the evaluation of diversity in multi-domain result sets must assess the ability of a given

algorithm to retrieve useful and diverse tuples within the first K results in the query answer. In this section we elaborate on the performance of the diversification algorithms described in Section 2 (`MMR`, `MaxSum`,`MaxMin`). We calculated a set of quantitative objective metrics, and we conducted a subjective user study. All the algorithms were evaluated using a $\lambda = 1$, thus giving equal importance to both diversity and ranking.

A well-known metrics for relevance in diversified result sets is the *α-Discounted Cumulative Gain* (α-DCG_K) [3], which measures the usefulness (gain) of a document based on its position in the result list and its novelty w.r.t. the previous results in the ranking. In the original formulation of α-DCG_K, documents are composed by a set of *information nuggets*. In the context of multi-domain result-sets, the α-DCG_K can be defined by assimilating an information nugget to a tuple $t_i \in R_i$, where $R_1, \ldots, R_n$ are the relations involved in a multi-domain query. Therefore, we define α-DCG_K as:

$$\alpha\text{-}DCG_K = \sum_{k=1}^{K} \frac{\sum_{i=1}^{n} J(\tau_k, t_i)(1-\alpha)^{r_{t_i,k-1}}}{\log_2(1+j)} \tag{19}$$

where $J(\tau_k, t_i)$ returns 1 when tuple t_i appears in a combination τ_k at position k in the result set and 0 otherwise. J is defined as $J(\tau_k, t_i) = \mathbb{1}_{\pi_{R_i}(\tau_k)=t_i}$, where π_{R_i} denotes the projection over the attributes of R_i, and $r_{t_i,k-1} = \sum_{j=1}^{k-1} J(\tau_j, t_i)$ quantifies the number of combinations up to position $k-1$ that contain the tuple t_i. In our experiments, α is set to 0.5 to evaluate novelty and relevance equally.

Sub-topic recall at rank K (S-$Recall_K$) [16] is a recall measure for search results related to several sub-topics, often applied to evaluate diversification algorithms [4]. In multi-domain search, a tuple in a combination can be assimilated to a subtopic in a document. Therefore, multi-domain recall at rank K (MD-$Recall_K$), defined next, measures, for each relation R_i involved in a query (with $i = 1 \ldots n$) and for all rank positions k from 1 to K, the set of *distinct* tuples ($R_i^k = \{t_i \in R_i | \exists j \leq k . \pi_{R_i}(\tau_j) = t_i\}$) retrieved in the result set, with respect to the entire population of the relation ($|R_i|$).

$$MD\text{-}Recall_K = \prod_{i=1}^{n} \frac{|\cup_{k=1}^{K} R_i^k|}{|R_i|} \tag{20}$$

3.1 Implementation and Datasets

We extended an existing architecture for multi-domain search application development [2] with a diversification component embedding the algorithms described in Section 2. Experiment has been performed on two usage scenarios. In the first scenario – Night Out *(NO)* – a user looks for a museum, a restaurant, and a hotel in Milan. We created a dataset consisting of the *Hotel* (50 tuples), *Restaurant* (50 tuples), and *Museum* (50 tuples) relations, where the initial 125,000 combinations have been pruned by removing all the triples for which the total walking distance from the Central Station in Milan is greater than 4 Km, which leaves

5000 combinations. We computed the mutual location distances, and we defined three quantitative relevance scores – the combination cost, the total walking distance, and the average ratings (see Example 1).

In the Study Abroad *(SA)* scenario, a user looks for a U.S. university, considering the rating of the university, the quality of life in the area, and the overall cost, including accommodation. The supporting dataset consists of three relations *University* (60 US universities with their academic quality score[1], walkability score of the surroundings[2], and average tuition fee), *State* (including their crime rate), and *Flat* (1200 flats). Joins were performed on the *state* attribute, yielding a dataset of 5100 combination. We defined three relevance scores: the overall yearly expenditure, a "quality" index[3], and the distance between the university and the flat.

3.2 Discussion

The evaluation covers both the *Night Out* and *Study Abroad* scenarios presented in Section 3.1. To avoid query-dependent bias, results are averaged over multiple experiments in each scenario. For the categorical case, 3 experiments have been performed, each of them applying one of the score functions described in Section 3.1, and diversification according to categorical distance. For the quantitative case, 6 experiments have been performed: for each score function in Section 3.1 the value computed by the remaining ones are, in turn, used to evaluate quantitative distance as in Equation 2.

Figure 2 shows α-DCG_K and MD-$Recall_K$ for the result sets obtained with no diversification and with the diversification algorithms `MMR`, `MaxSum`, and `MaxMin`, applying both categorical and quantitative distances. Each data point of the X-axis represents the k-th element in the result-set. The Y-axes represent, respectively, the values of α-DCG_K (Figure 2(a,b,e,f)) and of MD-$Recall_K$ (Figure 2(c,d,g,h)).

`MMR` and `MaxMin` always outperform the un-diversified baseline when used with the categorical diversity function both in terms of α-DCG_K and MD-$Recall_K$; `MaxSum` instead does not provide significant improvements with respect to the baseline. One can also notice that `MMR` and `MaxMin` offer similar performance: this is not surprising, as the greedy algorithms for the two objective functions are also very similar.

For quantitative distance, instead, all algorithms provide similar performance. In particular `MMR` and `MaxMin` degrade their performance with respect to categorical distance, and behave only slightly better than the baseline. This may also be influenced by the chosen quality measures (α-DCG_K and MD-$Recall_K$), that are based on diversity of extracted objects and therefore are more suited to evaluate

[1] Source: `http://archive.ics.uci.edu/ml/machine-learning-databases/university/`

[2] WalkScore - `http://www.walkscore.com/`

[3] A function of the academic quality score, the walkability index of a university and the crime rate in a state.

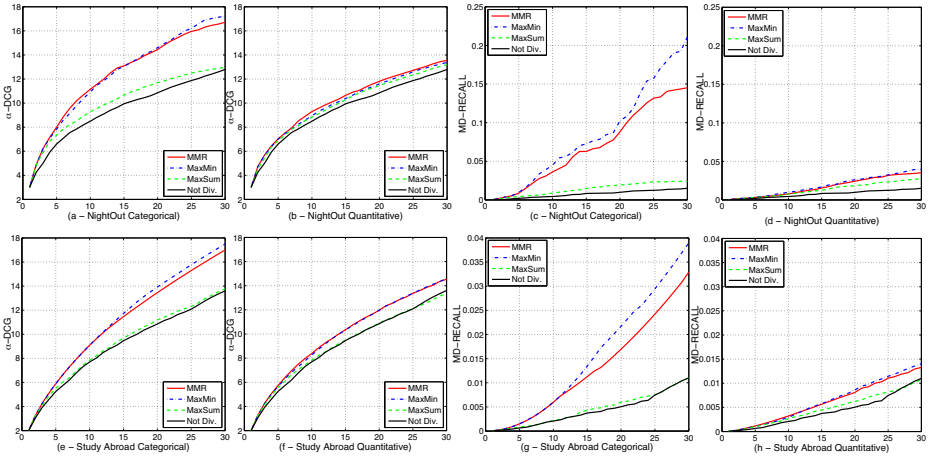

Fig. 2. Quantitative evaluation results. *Night Out* dataset: α-DCG_K for (a) categorical and (b) quantitative distances; MD-$Recall_K$ for (c) categorical and (d) quantitative distances. *Study Abroad* dataset: α-DCG_K for (e) categorical and (f) quantitative distances; MD-$Recall_K$ for (g) categorical and (h) quantitative distances.

categorical diversification. Nonetheless, we notice an overall coherent behavior of `MMR` and `MaxMin` in all settings; likewise, `MaxSum` consistently performs similarly to the un-diversified case. Overall, these results support the hypothesis that diversification algorithms can improve the quality of multi-domain result sets.

3.3 User Study

We conducted an (uncontrolled) user study focused on explicit comparison of result sets diversified according to the selected algorithms: users were asked to directly compare two alternative result sets (displaying 10 combinations each) and to select the one that, in their opinion, provided the best *quality* and *variety* of the items. As quantitative analysis provided evidence that `MMR` and `MaxMin` algorithms clearly outperformed the baseline when adopting categorical distance, we decided to consider only categorical distance for the user study; we also decided not to include `MaxMin` in the evaluation as its performance was comparable to `MMR`.

The study addressed both the scenarios *Night Out* and *Study Abroad* as described in Section 3.1. To avoid bias on the data instances, we generated 10 different subsets from the original result sets of each scenario, and then we applied separately the diversification algorithms to all of them. To avoid the effect of possible learning bias, the two scenarios were performed in random order; each user was shown two options among the three calculated result sets (un-diversified, `MMR` and `MaxSum`), in a round-robin fashion. The users could select their preferred result set in each scenario; they had unlimited time for completing the task. Each preference counted as a vote to the respective algorithm. The test was performed by 74 users, among which 25% were students and 75% were either search experts from industry or academia.

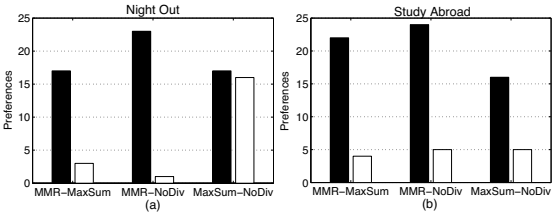

Fig. 3. *Direct comparison* user study: Preferences assigned to the different resultsets

Figure 3 shows the results of the voting. All the pairwise comparisons were subject to a binomial test, where the null hypothesis was that the preferences for both algorithms were equally likely to be expressed by the user. The perceived quality reflects quite well the quantitative results described in Section 3.2: result sets produced with the `MMR` algorithm were perceived to have higher quality and variety than both `MaxSum` and the un-diversified result (at significance level $\alpha = 0.01$). Conversely, the `MaxSum` algorithm was not significantly found better than the un-diversified result-set, as also suggested by the fact that the null hypothesis cannot be rejected.

The user experiment confirmed the considerations emerged from the quantitative evaluation, thus suggesting a user-perceivable benefit in the adoption of diversifications algorithm in multi-domain search applications.

4 Related Work

The evolution of search systems towards the extraction of structured information from Web content have been widely addressed in several recent works (e.g. *Concept search* [7]). Multi-domain search [2] focuses on processing queries involving several topics and domains on Web data sources. The present work explores diversification in this context, as a mean for improving the utility of result sets made of associated entity instances.

Result diversification is a well-investigated topic; [6] provides a survey of existing approaches, while [8] discusses a systematic axiomatization of the problem, that is the base of the formalization of multi-domain diversification in Section 2. A broad distinction can be done between the contributions that focus on diversifying search results for document collections (e.g. [12]) and those that concentrate instead on structured data sets [10,13,15].

Our work is mostly related to diversification applied to structured data. In this field, diversification in multiple dimensions is addressed in [5], where the problem is reduced to MMR by collapsing diversity dimensions in one composite similarity function. The work in [15] examines the diversification of structured results sets as produced by queries in online shopping applications. The paper shows how to solve *exactly* the problem of picking K out of N products so to minimize an attribute-based notion of similarity and discusses an efficient implementation technique based on tree traversal. Multi-domain diversification, as

discussed in this paper, is a broader problem; it could be partially reduced to prefix-based diversification only in the case of categorical diversity, by choosing an arbitrary order for the categorical attributes used to measure combination diversity. Keyword search in structured databases is addressed in [4], where diversification is not applied to result sets, but to query interpretations, which are assumed to be available from the knowledge of the database content and query logs. The multi-domain search applications addressed in this paper assume for simplicity unambiguous queries and thus a fixed interpretation, but could reuse the interpretation diversification approach of [4] to cope for multi-domain searches with more than one possible interpretation. A recent related work is [13], which applies to the selection of Web services characterized by their non-functional properties. The authors introduced a novel diversification objective, `MaxCov`, which leads to the selection of items with high relevance score, such that the remaining ones are not too far from any of the elements of the diversified result set. We plan the testing of `MaxCov` as part of the future work.

Finally, the work [10] investigates the diversification of structured data from a different perspective: the selection of a limited number of features that can maximally highlight the differences among multiple result sets. Although the problem is apparently different from multi-domain search (the actual goal of [10] is to find a set of attribute values that maximally differentiates a number of input results set, respecting a size upper bound) identifying the best attributes to use for ranking and diversification is relevant to multi-domain search as well, and we have started addressing it by studying how the distribution of attribute values affects the capability of the ranking function to sample the population of the input relations evenly.

5 Conclusions

Multi-domain search is a promising trend in search applications; however, to preserve the current ability of search engines to squeeze in one page the most interesting results, the combinatorial explosion of result sets formed by several correlated entity instances must be tamed. In this paper, we have investigated the problem of multi-domain result set diversification, by showing how the diversification techniques well studied in the context of IR can be extended to support this class of applications. We experimentally tested three algorithms for the trade-off between relevance and diversity, and showing that they can introduced a significant degree of diversification in the result set; a user study demonstrated a positive perception of the utility of diversification by users.

In the future, we plan to extend this work in several directions. On the methodological side, we plan to better investigate the interplay between the score function and the similarity measure (beyond the simple case of uniform data distribution studied in Section 2), so to propose a methodology for the selection of the most promising scoring and diversity functions. On the architecture side, we will investigate issues like the design of appropriate data and index structures for efficient diversification, relevance and diversity-aware caching of

results, and the thorough evaluation of the overhead of diversification. Finally, we plan a careful graphical user interface design and a novel round of user testing of the multi-domain search concept, this time using online data and real users.

Acknowledgments. This research is partially supported by the Search Computing (SeCo) project, funded by European Research Council, under the IDEAS Advanced Grants program; by the Cubrik Project, an IP funded within the EC 7FP; and by the BPM4People SME Capacities project. We wish to thank all the participants to the user study and all the projects contributors.

References

1. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336. ACM, New York (1998)
2. Ceri, S., Brambilla, M.: Search Computing Systems. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 1–6. Springer, Heidelberg (2010)
3. Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 659–666. ACM, New York (2008)
4. Demidova, E., Fankhauser, P., Zhou, X., Nejdl, W.: Divq: diversification for keyword search over structured databases. In: SIGIR 2010: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 331–338. ACM, New York (2010)
5. Dou, Z., Hu, S., Chen, K., Song, R., Wen, J.-R.: Multi-dimensional search result diversification. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 475–484. ACM, New York (2011)
6. Drosou, M., Pitoura, E.: Search result diversification. SIGMOD Rec. 39(1), 41–47 (2010)
7. Giunchiglia, F., Kharkevich, U., Zaihrayeu, I.: Concept search: Semantics enabled syntactic search. In: SemSearch, pp. 109–123 (2008)
8. Gollapudi, S., Sharma, A.: An axiomatic approach for result diversification. In: WWW 2009: Proceedings of the 18th International Conference on World Wide Web, pp. 381–390. ACM, New York (2009)
9. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoretical Computer Science 38, 293–306 (1985)
10. Liu, Z., Sun, P., Chen, Y.: Structured search result differentiation. Proc. VLDB Endow. 2(1), 313–324 (2009)
11. Martinenghi, D., Tagliasacchi, M.: Proximity rank join. PVLDB 3(1), 352–363 (2010)
12. Rafiei, D., Bharat, K., Shukla, A.: Diversifying web search results. In: WWW 2010: Proceedings of the 19th International Conference on World Wide Web, pp. 781–790. ACM, New York (2010)
13. Skoutas, D., Alrifai, M., Nejdl, W.: Re-ranking web service search results under diverse user preferences. In: PersDB 2010 (September 2010)

14. Soliman, M.A., Ilyas, I.F., Saleeb, M.: Building ranked mashups of unstructured sources with uncertain information. PVLDB 3(1), 826–837 (2010)
15. Vee, E., Srivastava, U., Shanmugasundaram, J., Bhat, P., Yahia, S.A.: Efficient computation of diverse query results. In: ICDE 2008: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, pp. 228–236. IEEE Computer Society, Washington, DC (2008)
16. Zhai, C.X., Cohen, W.W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 10–17. ACM, New York (2003)