Federated Entity Search Using On-the-Fly Consolidation

Daniel M. Herzig[1,⋆], Peter Mika[2], Roi Blanco[2], and Thanh Tran[1]

[1] Karlsruhe Institute of Technology (KIT), Germany
[2] Yahoo! Research, Spain
{herzig,ducthanh.tran}@kit.edu, {roi,pmika}@yahoo-inc.com

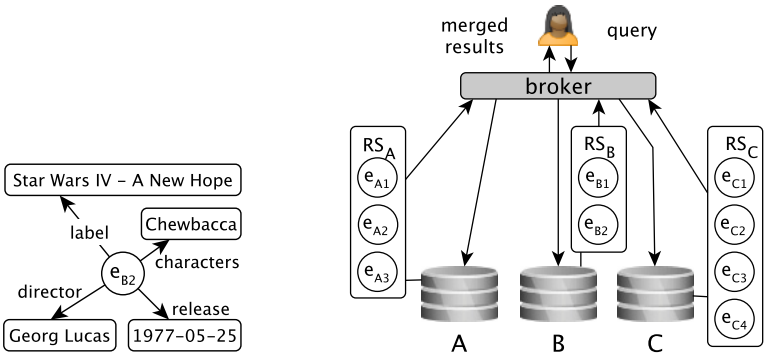
Abstract. Nowadays, search on the Web goes beyond the retrieval of textual Web sites and increasingly takes advantage of the growing amount of structured data. Of particular interest is entity search, where the units of retrieval are structured entities instead of textual documents. These entities reside in different sources, which may provide only limited information about their content and are therefore called "uncooperative". Further, these sources capture complementary but also redundant information about entities. In this environment of uncooperative data sources, we study the problem of *federated entity search*, where redundant information about entities is reduced on-the-fly through *entity consolidation* performed at query time. We propose a novel method for entity consolidation that is based on using language models and completely unsupervised, hence more suitable for this on-the-fly uncooperative setting than state-of-the-art methods that require training data. Further, we apply the same language model technique to deal with the federated search problem of ranking results returned from different sources. Particular novel are the mechanisms we propose to incorporate consolidation results into this ranking. We perform experiments using real Web queries and data sources. Our experiments show that our approach for federated entity search with on-the-fly consolidation improves upon the performance of a state-of-the-art preference aggregation baseline and also benefits from consolidation.

1 Introduction

Taking advantage of the growing amount of structured data on the Web has been recognized as a promising way to improve the effectiveness of search and has therefore gained the interest of researchers and industry [1]. This development is also driven by the demand from Web search users, whose most dominant search task is the search for entities. Recent studies showed that about 70% of Web search queries contain entities [2] and that the intent of about 40% of unique Web queries is to find a particular entity [3]. In contrast to named entities, which are text tokens identifying specific concepts, e.g. the name of a person, the increasing amount of structured data on the Web as well as the availability of knowledge bases allows to perceive entities not just as single tokens, but as structured objects with attributes and values, e.g. Figure 1a illustrates an entity representing the movie "Star WarsIV - A New Hope".

⋆ Work done while being visiting researcher at Yahoo! Labs, Barcelona.

H. Alani et al. (Eds.): ISWC 2013, Part I, LNCS 8218, pp. 167–183, 2013.



(a) A structured entity with four attributes and values

(b) The broker obtains only the ranked result lists RS_A, RS_B, RS_C from each data source and merges them into one single ranked list

Fig. 1. Federated Entity Search in an uncooperative setting

The entities reside in different sources across the Web or originate from different knowledge bases. These sources capture redundant but also complementary information about entities. Hence, consolidating co-referent entities referring to the same real-world object and providing *search functionalities over co-referent entities* is a crucial step towards exploiting structured data sources for retrieval. In particular, the need for large scale and fast coreference has been recognized and recently a solution in this direction has been proposed [4]. This solution and the comprehensive work of the database community in this realm [5–7] assume full access to the entire datasets to compute features such as weights of attributes, co-occurences or to learn parameters, which are then used to resolve all coreferences between two or more datasets in one run. However, access to the entire datasets is either not granted in many application scenarios such as search over multiple Web data sources (where data access is only provided via APIs for single requests), also called federated search over *uncooperative* sources [8, 9], or many data sources are highly dynamic, imposing a high burden on batch processing to keep up with frequent changes and to provide fresh information for time sensitive applications such as search over stock quotes, movies and timetables. Distributed document retrieval for uncooperative environments has been studied in the IR community [8, 9]. We investigate the task of federated entity search with structured entities consisting of a varying number of attributes and corresponding values. This task is different from document retrieval, where each document consists of exactly one text body.

Contributions. We address the setting of *uncooperative environments*, where neither prior information about the data sources nor training data is available and propose a language model (LM) based approach for *federated entity search in uncooperative environments* using query time entity consolidation.

We present three contributions: (1) We propose a LM based unsupervised approach for computing the similarity between entities and use it to perform query time entity consolidation. (2) We reuse these LM based representations of entities and we show how this entity representation in combination with composite relevance models [10, 11] can be used to obtain a combined ranking of results returned from different sources. The mechanisms we propose to incorporate consolidation results into this ranking are particularly novel. (3) In our experiments, we employ real-world Web queries and data sources and investigate the effects of federated search in combination with consolidation on retrieval performance. We show that our approach exceeds a state-of-the-art preference aggregation method for federated search [12] and show the advantages of consolidation for search in federated settings.

2 Overview

We follow the definition by Pound et al. [3] and define *entity search* as the task of answering arbitrary information needs related to particular aspects of entities, expressed in unconstrained natural language and resolved using a collection of structured data. We address a particular kind of entity search, namely the search over multiple data sources, called *federated search*, which entails the three main problems of *source representation*, *source selection*, and *result merging* [8, 9]. We focus on the latter for *federated entity search in uncooperative settings* as illustrated in Figure 1b, where only ranked result lists of entity descriptions are obtained from each source and no further information about the sources is available. In this scenario, we perform *consolidated entity search* where entities representing the same real-world object, called *co-referent entities* are identified, linked and incorporated into ranking to avoid redundant results. Further, Web data is heterogeneous in the sense that differences in schema and vocabulary are common. Coping with schema differences has been studied in the area of schema matching and for the given setting also in our previous work [11]. Here, it is considered as out of scope. We illustrate the addressed problem throughout the paper using the following example.

Example 1. Assume the keyword query *"star wars"* is issued to three data sources, which hold information about movies. Each source returns a ranked list of structured entities as illustrated in Figure 1b and the same lists are shown in more detail in Figure 2a. We observe that the lists contain co-referent entities, e.g. e_{B2} in RS_B and e_{C1} in RS_C both represent the same movie "A New Hope". If we put all entities in one result list, we obtain a list with redundant results. Later, we will refer to this case **w**ithout consolidation as CRM_w. In our example, we observe in Figure 2b that within the first six ranks only three distinct entities are shown, because $\{e_{B2}, e_{C3}, e_{A2}\}$ and $\{e_{C3}, e_{C2}\}$ are co-referent (indicated by identical line type). Our goal is to **c**onsolidate the results by grouping co-referent entities into sets as illustrated in Figure 2c, which we will later refer to as CRM_c. In particular, we will focus on ranking in this setting and investigate the effects on retrieval performance of federated search with and without consolidation.

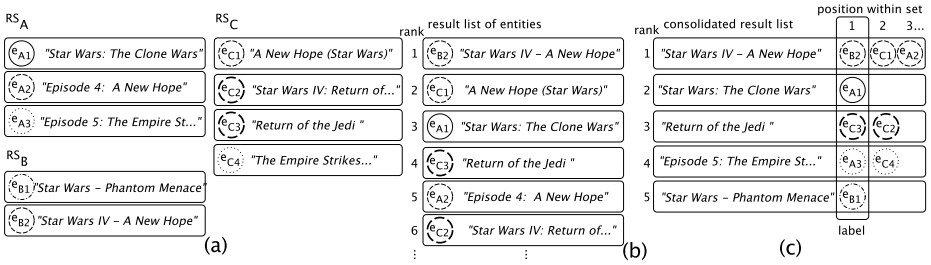


Fig. 2. (a) Result lists of each source. (b) Merged result list containing co-references without consolidation (CRM_w). (c) Consolidated result list with co-referent sets using the entity on position 1 as label (CRM_c).

We focus on Web data for which the RDF model has been proposed as a W3C standard for data representation and interchange. For the sake of generality, we omit RDF specific features, like blank nodes, and employ a general graph-structured data model.

A *data source* is a directed and labeled graph $G = (N, E)$. The set of nodes N is a disjoint union of *entities* N_E and literals N_L, i.e. $N = N_E \uplus N_L$. Edges E can be conceived as a disjoint union $E = E_E \uplus E_L$ of edges representing connections between entities, i.e. $a(e_i, e_j) \in E_E$, iff $e_i, e_j \in N_E$, and connections between entities and literals also called *attribute values*, i.e. $a(e, v) \in E_L$, iff $e \in N_E$ and $v \in N_L$. Given this graph G, we call the bag of attribute value edges $A(e) = \{a(e, v) \in E | v \in N_L\}$ the *description* of the entity $e \in N_E$, and each $a(e, v) \in A(e)$ is called an *attribute* of e. The set of distinct attribute labels of an entity e, i.e. $A'(e) = \{a | a(e, v) \in A(e)\}$, is called the *model* of e. Figure 1a illustrates entity e_{B2}, which has the model $A'(e_{B2}) = \{label, director, characters, release\}$. In our Web scenario, each data source is represented by a graph G_X, for example Figure 1b illustrates three data sources $X = \{A, B, C\}$. Although arbitrary edges can connect entities across data graphs, we are only interested in edges denoting that two entities are co-referent, i.e. $a_{\text{same}}(e_X, e_Y), e_X \in G_X, e_Y \in G_Y$, e.g. `owl:sameAs` [1].

3 On-the-fly Entity Consolidation

Entity consolidation is typically performed through the main steps of *representing entities* as attribute value pairs, and finding the appropriate *similarity metric* and *threshold* to determine whether two given entity representations refer to the same object or not, i.e. when the similarity computed using the metric exceeds the threshold. Since several attributes are typically used, state-of-the-art methods employ supervised machine learning techniques to learn the weights for attributes or also the metrics and thresholds [13]. In this section, we (1) represent attribute values as language models (LMs), (2) employ a specific notion

[1] `http://www.w3.org/TR/owl-ref/#sameAs-def`

of distance for LMs as the similarity metric, and (3) propose an unsupervised technique to estimate the weight associated with each attribute LM. In our approach, all the steps needed to derive the LM-based entity representation as well as the actual detection of coreferences are performed on-the-fly during the execution of a query.

3.1 Entity Representation

In entity search, composite LMs have been proposed to represent an entity as a collection of multinomial distributions, each capturing one particular entity attribute [11, 14] and are used to compute the similarity between an entity and a query. This modeling is suited when entities are not only associated with concise values but descriptions – which is the case for many Web data sources capturing entities via attributes such as *label* and *comment*. We use this representation also for the consolidation task.

The composite LM $\mathcal{P}$ for an entity e is constructed by considering all attributes in the model of e, $a \in A'(e)$. That is, $\mathcal{P}$ contains a LM $P_e(w|a)$ for every $a \in A'$. Every $P_e(w|a)$ captures the probability of observing a word w in the attribute values of a associated with e. Let $V_a(e)$ be the bag of value nodes of the attribute a associated with e, i.e. $V_a(e) = \{v|a(e,v) \in E, v \in N_L\}$, and w a word in the vocabulary, then $P_e(w|a)$ is estimated using maximum-likelihood as follows:

$$P_e(w|a) = \frac{\sum_{v \in V_a(e)} n(w,v)}{\sum_{v \in V_a(e)} |v|} \tag{1}$$

where $n(w,v)$ denotes the count of w in the value v, and $|v|$ is the total number of words in v.

3.2 Similarity Metric

Given two entities in the result lists, $e_X \in RS_X$ and $e_Y \in RS_Y$, we determine whether they are co-referent or not using a similarity metric. Standard metrics used by consolidation methods include *edit distance* and *Jaccard similarity*, which can be applied to two given attribute values. The former captures the number of edit operations needed to transform one value to the other while the latter is based on the word overlaps between the two values. Since we apply LMs to captures values, we measure the overlap of the LMs with the *Jensen-Shannon divergence* (JSD), which is based on the *Kullback-Leibler divergence* (KLD). The JSD however has the advantages of being symmetric, bounded ($0 \leq JSD \leq 1$), smoothed and its square root is a metric. Given the probability distributions P_X, P_Y and $R = \frac{1}{2}P_X + \frac{1}{2}P_Y$, the JSD is defined as:

$$JSD(P_X||P_Y) = \frac{1}{2}KLD(P_X||R) + \frac{1}{2}KLD(P_Y||R) \tag{2}$$

where $KLD(P||R) = \sum_w P(w) \log_2 \frac{P(w)}{R(w)}$. For computing the distance d between two entities e_X and e_Y, $d(e_X, e_Y)$, we use the square root of the JSDs calculated

over the LMs constructed for all attributes a that both entities have in common and weight each overlap measured by the JSD by $\omega(a)$:

$$d(e_X, e_Y) = \frac{1}{\sum_{a \in A'(e_X) \cap A'(e_Y)} \omega(a)} \sum \omega(a)\ JSD(P_{e_X}(w|a)||P_{e_Y}(w|a))^{\frac{1}{2}} \tag{3}$$

3.3 Estimating Weights

The weight $\omega(a)$ expresses how discriminative and identifying an attribute a is. We determine $\omega(a)$ w.r.t. the result lists RS_X and RS_Y. First, we construct a LM $P_X(w|a)$ analog to Equation 1. However, $P_X(w|a)$ captures the values of all the entities in RS_X instead of a single entity, i.e. $V_a(RS_X) = \{v|a(e_X, v), e_X \in RS_X\}$ instead of $V_a(e)$. Then, we compute the entropy $H(P) = -\sum_w P(w) \log_2 P(w)$ and set $\omega(a)$ to:

$$\omega(a) = \frac{1}{2} H(P_X) H(P_Y) \tag{4}$$

The rationale behind this formulation is that the entropy is high, if the bag $V_a(RS_X)$ contains many diverse values, and it is low, if $V_a(RS_X)$ contains similar and hence less discriminative values. Attributes with more diverse values are associated with higher weights because they provide more discriminate information to distinguish entities.

3.4 Entity Similarity

Given the above distance function and two result lists $RS_X = (e_{X1}, e_{X2},..., e_{Xi},...)$ and $RS_Y = (e_{Y1}, e_{Y2},..., e_{Yj},...)$, we consider two entities $e_X \in RS_X$ and $e_Y \in RS_Y$ as co-referent, if their distance is below a threshold t and if they are mutually the closest to each other, see Equation 5. The latter condition assures that only co-references are established, if a candidate entity is favored over all alternatives. Note, that we also compare the sources to themselves, i.e. $X = Y$, to find co-references within a source.

$$\begin{aligned} &d(e_X, e_Y) < t \quad \wedge \\ &d(e_X, e_Y) = \min_i d(e_{Xi}, e_Y) = \min_j d(e_X, e_{Yj}) \end{aligned} \tag{5}$$

4 Ranking Consolidated Entities

We continue our previous Example 1 to illustrate our procedure. First, we obtain the result lists RS_X for each source X as depicted in Figure 2a. In addition, we have now a set of edges A_{same} linking co-referent entities. Figure 3 depicts this situation for our example, we see the three result lists and four a_{same} edges (arrows). The co-references A_{same} are either obtained through our consolidation process, are already part of the data or provided by external services such as `http://sameas.org`. We aim at merging these entities into one ranked list while taking the coreferences into account. In the following we present our ranking model for consolidated entities and then show our strategy for exploiting co-references.

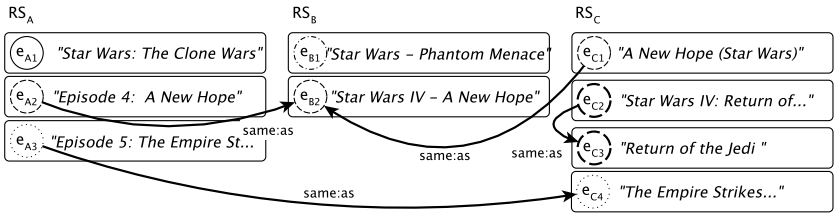


Fig. 3. Three lists RS_X with four $a_{\texttt{same:as}}$ edges (arrows)

4.1 Ranking for Structured Web Data

The general concept we apply for ranking is based on pseudo-relevance feedback [10]. We adapt this idea and apply it to federated entity search. In line with this concept, we build two models, one Query Model (QM) capturing the information need with the help of relevance feedback and Resource Models (RM) representing results to be ranked. Each RM is scored against QM and sorted by its score into the final result list. Both models, QM and RM, share in general the same structure as entities, as described in Section 3.1, i.e. they contain a set of attribute labels A' and a corresponding LM $P \in \mathcal{P}$ for each attribute. Formally, the model $M \in \{QM, RM\}$ is a 3-tuple $M = (\mathcal{E}, A', \mathcal{P})$. We denote the set of entities $\mathcal{E}$ of model M as $\mathcal{E}(M)$ and the set of attributes A' of model M as $A'(M) = \{a | a \in A'(e), \exists e \in \mathcal{E}(M)\}$. As before for the consolidation, we use the JSD (Equation 2) to measure the distance between two corresponding LMs for all attributes that QM and RM have in common:

$$Score(QM||RM) = \sum_{a \in \bigcap_X A'(RS_X)} JSD(P_{QM}(w|a)||P_{RM}(w|a))^{\frac{1}{2}} \tag{6}$$

The LMs of QM and RM, see Equation 7, are computed from the respective LMs of the entities that are comprised by the model and each entity LM P_e (Equation 1) is weighted with an entity specific weight $\mu(e)$:

$$P_M(w|a) = \frac{\sum_{e \in \mathcal{E}(M)} \mu(e) P_e(w|a)}{\sum_{e \in \mathcal{E}(M)} \mu(e)} \tag{7}$$

The weight μ is the crucial part of the query model QM. For RM the weight is constant $\mu = 1$. The weight allows to control the impact of each entity on the query model. With the weight μ, we adapt the ranking framework to the federated search setting and exploit the ranking of the individual sources. We use the discounted rank $r(e_X)$ of the entity e_X in the result list RS_X to weight its influence on the query model:

$$\mu(e_X) = \frac{1}{\log(1 + r(e_X))} \tag{8}$$

By using the ranks in μ, we take advantage of the ranking of the sources. Although the sources do not provide any information explicitly about themselves, all their knowledge, such as domain expertise, popularity, click-data, and

other signals, are incorporated in their ranking function and thereby implicitly conveyed in the ranking. Moreover, we tie the importance of an entity represented by its rank to the content and the structure of the entity, which is captured by the LM of the entity. For QM, we use all entities returned by the sources, i.e. $\mathcal{E}(QM) = \bigcup_X RS_X$ and we construct one RM for each entity. Note that an advantage of the above technique is that it is entirely parameter free. The whole ranking procedure takes all its ingredients from the results returned by the sources for the initial query. This is an important feature for dynamic web environments, where data sources may (dis-)appear frequently and a prior integration into the federated search process is not possible.

4.2 Ranking Consolidation

Given the result lists RS_X, we consider each entity $e \in \bigcup_X RS_X$ individually and construct a corresponding model RM for each entity. Then, we compute a score for each RM and sort each entity by its score to obtain a ranked list of entities. This ranked list contains all entities in $\bigcup_X RS_X$, one entity on each rank as depicted in Figure 2(b). At this point we have a ranked list without taking advantage of the co-references. In the experimental Section 5, we refer to this stage as CRM_w. Now, we allow sets of entities on each rank instead of a single entity. We iterate through the ranked list from the best to the last ranked entity. During this iteration we make use of the co-references A_{same}. If we observe an entity that has co-references, we position the set of all co-referent entities on this rank and remove them from their original ranks. Within each rank, the entity previously ranked highest is first and then we order the co-referent entities by their previous ranks, see Figure 2(c). The result of this strategy is a list of ranked sets. In the next section, we refer to this consolidated ranking strategy as CRM_c.

5 Experiments

We conducted experiments on consolidation and on ranking in two real-world scenarios. In one scenario users search for movies and in the another one for scientific publications. We used publicly accessible APIs available on the Web as sources of entities. Table 1 lists the sources for both scenarios (RT is abbr. for rottentomatoes.com and MS for Microsoft Academic Search). We used Yahoo! Dapper[2] to mimic an API using the site search of Citeseer and ACM. All data used in the experiments is available at `http://www.aifb.kit.edu/web/Dhe/data`.

Real-World Web Search Queries. We extracted 50 real Web search queries for each scenario from a Web search engine query log. For each scenario, we manually created a list of more than ten hostnames, which contains those of the data sources and highly popular sites. We sampled only queries having at least two clicks on one of these hostnames to obtain queries for our scenarios. Details on the query sets and the obtained result lists are given in Table 1 and the first six queries of each set are shown in Table 2.

[2] `www.open.dapper.net`

Table 1. Queries and obtained result lists RS

	Source	#q	$\|q\| \pm \sigma$	max$\|RS\|$	$avg\|RS\| \pm \sigma$	$\|RS\|=\emptyset$
Movie	MovieDb	50	2.58±1.3	20	6.18 ± 7.22	6
	Netflix	50	2.58±1.3	100	81.7 ± 30.4	0
	RT	50	2.58±1.3	50	12.0 ± 15.5	0
Publ.	Arxiv	50	4.4±2.1	100	83.2±36.2	0
	ACM	50	4.4±2.1	20	18.6±4.21	0
	Citeseer	50	4.4±2.1	10	9.2±2.51	3
	MS	50	4.4±2.1	100	89.0±27.3	0

Table 2. Queries with *movie*-related intend (left) and *scientific* intend (right)

mission impossible 4	parameter selection in particle swarm optimization
the debt	mobility models in inter-vehicle communications literature
hobbit	computer effective to academic learning
cowboys and aliens 2011	bivariate f distribution
the hunters 2011	werner krandick
star wars	using truth tables to evaluate arguments

Ground Truth. We obtained the ground truth for both tasks through expert judgments. The distributions of the co-references between the sources are given in Table 3. We can observe that co-references exist not just between but also within the results of a data source. Noteworthy, one source (MS) is dominant in the publication scenario and is part of 83% of the co-references. We followed the methodology of [15] to obtain relevance judgments for the ranking evaluation. We rated the top-10 results for each query. In total, there are 604 relevant entities for the movie scenario, which are distributed among the sources as follows: RT: 40%, Netflix: 36%, MovieDb: 23%. For the publication scenario, the raters judged 997 entities as relevant. The distribution of the relevant results is here highly skewed. MS returned 53% of the relevant results, ACM 24%, Arxiv 15%, and Citeseer 8%. Details on the ground truth are shown in Table 4, such as the number of raters and in particular the inter-rater agreement measured on the "Overlap" with Krippendorff's α for ordinal values [16]. Overall, we consider the agreement of $\alpha_{ordinal} > 0.66$ high enough to rely on the ground truth [17].

Table 3. Ground truth co-references

	MovieDb	RT	Netflix
MovieDb	4		
RT	179	33	
Netflix	162	248	10
		Total:	636

	ACM	Arxiv	Citeseer	MS
ACM	25			
Arxiv	14	62		
Citeseer	13	1	5	
MS	239	75	59	232
			Total:	725

5.1 Consolidation Results

The main focus of our work is on the ranking of consolidated entities. In order to obtain co-references, we applied the procedure described in Section 3. In Figure 4 we see the effect of the threshold t, the only parameter necessary in our approach, on the the metrics *F1*, *Precision*, and *Recall* w.r.t to coreferences between sources. As we have seen in the previous section, many co-references exist also within one data source. We can assume that within a source the same vocabulary is used and therefore entities are inherently closer to each other in terms of our similarity metric compared to entities of different sources. Hence, a lower threshold is needed when consolidating entities of the same source. As a consequence, we reduce t by 0.2 in this case and report evaluation results for $t_{\text{movie}} = 0.7$ and $t_{\text{pub}} = 0.6$ for the respective scenario. We evaluate consolidation from two perspectives. First, we look at each single co-reference link and second, we evaluate the entire co-reference sets created from these links. Each co-reference is classified as true/false positive/negative (abbr. TP, FP, TN,FN). In Table 5c and 5d we see the confusion matrix for both scenarios over the entire query set. The consolidation performance as an average of the co-references created for each query is reported in Table 5a and the corresponding numbers for the sets of co-references are shown in Table 5b. Overall, the performance numbers are high. Although a direct comparison is not possible, the numbers are in the same order of magnitudes as previously reported for supervised consolidation [18]. We now apply these co-references for consolidated retrieval in the next section.

Table 4. Ground truth statistics and agreement α

	Raters	Subjects	Ratings	Overlap	$\alpha_{ordinal}$
Consolidation					
Publications	6	3076	4246	1170	**0.7596**
Movie	3	5783	6061	278	**0.8204**
Relevance					
Publications	6	2736	3022	286	**0.7051**
Movie	3	1616	1992	376	**0.6919**

Table 5. Consolidation performance

Avg. per Q	Movie	Publ.
F1-score	0.8233	0.7672
Accuracy	0.9982	0.9996
Precision	0.8063	0.8636
Recall	0.8781	0.7118

(a) Coreferences

Avg. per Q	Movie	Publ.
#Co-ref Sets	6.9799	9.7000
Set size	2.5100	2.0740
Set Purity	0.7737	0.8713

(b) Sets of coreferences

Movies	
TP:556	FP:166
FN:80	TN:286571

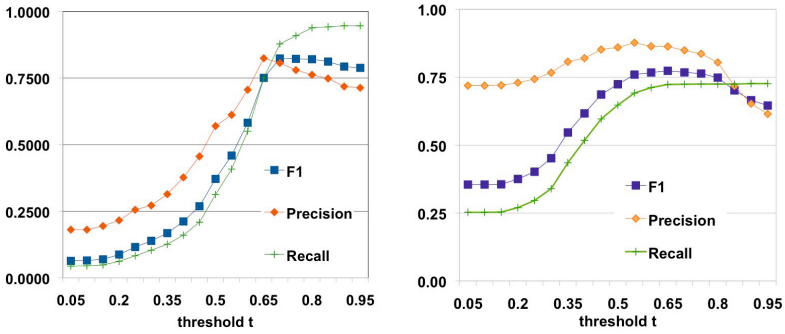
(c) Confusion matrix

Publications	
TP:518	FP:77
FN:207	TN:1049954

(d) Confusion matrix

5.2 Ranking Evaluation

We aim to answer two questions: What is the effect of federated search on retrieval compared to the performance of the sources separately? Does consolidation improve federated search? We assess the ranking using Normalized Discounted Cumulative Gain (*NDCG*) and report the results in Table 6a for the movie and in Table 6b for the publication scenario and indicate statistically significant improvements using Fisher's two-sided, paired randomization test [19].



(a) movie (b) publication

Fig. 4. Consolidation over threshold t

Systems. We implement our *Consolidated Relevance Model (CRM)* approach in two different ways. (1) We employ a federated version ***without*** using co-references (CRM_w) and (2) a federated and ***consolidated*** version exploiting co-references (CRM_c) as described in Section 4. We compare CRM against two baselines, the individual rankings of the sources and the *Multinomial Preference Model (MPM)*, a state-of-the-art rank aggregation strategy [12]. We use an unsupervised version of MPM, i.e. without the supervised adherence parameter, and study two preference encodings. The first encoding $C(e_i, e_j)$ is binary (labeled *MPM*), where one entity e_i is preferred over entity e_j when e_i has a lower rank ($r(e)$ denotes the rank of e in the result list RS):

$$C(e_i, e_j) = \begin{cases} 1 \text{ if } r(e_i) < r(e_j) \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

The second encoding exploits the difference between ranks to express the degree of how much e_i is preferred over e_j using discounted ranks (labeled with subscript d as MPM_d):

$$C_d(e_i, e_j) = \begin{cases} \frac{1}{\log(1+r(e_i))} - \frac{1}{\log(1+r(e_j))} & \text{if } r(e_i) < r(e_j) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Evaluation Settings. Note that all systems return a ranked list of individual entities except for those that make use of consolidation, where each result in the

ranked list returned by MPM, MPM_d and CRM_c represents a set of entities instead of a single entity as illustrated in Figure 2(c). In order to assess the relevance of such a set, we use the best ranked entity within that set as the representative element, called the label as depicted in Figure 2(c). The relevance of the set is determined based on the relevance of its label, except for *expand* as described below. We evaluate the systems in three different settings:

Table 6. Retrieval performance (NDCG)

System	Std	NRel	Expand
MovieDb	0.4128	0.4025	n/a
RT	**0.5360**	**0.5165**	n/a
Netflix	0.5191	0.5141	n/a
CRM_w	**0.8699**†	**0.4992**	n/a
MPM	0.4936	0.5037	0.8070
MPM_d	0.5232	0.5232	0.8366
CRM_c	**0.5787**†*	**0.5515**$^{\circ\bullet}$	**0.8744**$^{\circ}$

†stat. diff. $\alpha < 0.05$ to RT, *to MPM, MPM_d
$^{\circ}$stat. diff. $\alpha < 0.05$ to MPM, $^{\bullet}$to CRM_w

(a) Movie scenario results (NDCG)

System	Std	NRel	Expand
Arxiv	0.1824	0.1737	n/a
ACM	**0.3537**	**0.3455**	n/a
Citeseer	0.1630	0.1551	n/a
Publ. 3 sources (Arxiv, ACM, Citeseer)			
CRM_w	**0.3592**	**0.3310**	n/a
MPM	0.2273	0.2273	0.2434
MPM_d	0.2541	0.2542	0.2697
CRM_c	**0.3524***	**0.3462**$^{\star\bullet}$	**0.3697***
Publ. 4 sources (as above and MS)			
MS	**0.6474**	**0.5976**	n/a
CRM_w	**0.5463**	**0.4430**	n/a
MPM	0.4568	0.4230	0.4894
MPM_d	0.4869	0.4743	0.5291
CRM_c	**0.5096**	**0.4822***	**0.5604**

*stat. diff. $\alpha < 0.01$ to CRM_w

(b) Publication scenario results (NDCG)

Std: First, we assess the results in the *standard* way by going through the ranked lists as they are returned by the systems and simply assess the relevance of each rank using the ground truth.

Nrel: In the Std. setting, results are considered relevant even if the same results (i.e. co-referent entities) have been seen in the list before. The *Nrel* setting accounts for redundancy by considering subsequent occurrences of co-referent entities as *non-relevant*, as suggested by [18]. Even if a result is relevant according to the ground truth, it is considered here as not relevant when a co-reference has already been seen.

Expand: The third setting gives special treatment to the systems MPM, MPM_d and CRM_c that perform consolidation. In the previous settings, relevance assessment of these systems is simply based on the labels of result sets. In this *expand* setting, we assess the results in the way proposed for clustered IR [20]. The idea is that a user goes from the top to the bottom of the result list, and checks the label of each cluster (set of results in this case). If a label is considered

relevant, the set is expanded and each entity in the set is assessed individually using the ground truth.

Std Results. First, we look at the effect of federation. When comparing the single sources, shown in the first three lines in Table 6a for the movie scenario, we observe that RT performs best (bold digits). Further, note that the performance differences among these sources are relatively small. The federated approach CRM_w outperforms the individual results by 62%. For the publ. scenario, we look at two setups, one with the three sources that share about the same amount of co-references and relevant results, and a second setup with a fourth source - MS, which is 'an outlier' because it contains 53% of the relevant results and is part of 83% of the co-references, as described above. For the 3-source setup, we observe a different initial situation, see Table 6b. The best source ACM performs about twice as good as the second best source Arxiv. Given this unbalanced situation, our federation approach CRM_w performs only marginally better than the best source. When we look at the publ. scenario with 4 sources in the lower part of this table, we observe that the source MS is again twice as good as the previously best source ACM. Further, adding MS to the pool of sources, improves the performance of CRM_w. However in this skewed setting, the federated approach CRM_w performs not as good as the best source MS, but better than the three other sources. In summary for the std. setting, we observe that federation improves retrieval if the sources have about the same performances. Otherwise, it yields improvements upon most sources but cannot guarantee the best performance. Next, we investigate the effect of federation in combination with consolidation, i.e. the systems MPM, MPM_d and CRM_c. Through consolidation entities are grouped into sets and as a consequence there are less (relevant) results, i.e. (relevant) set labels, in the ranked list after consolidation than entities in the list before consolidation. In the movie case, where more co-reference sets exist (3.6 sets per query in the top10 ranks), we observe as expected that NDCG is much lower than without consolidation. The same holds for the publ. scenario although the difference is smaller because there are fewer and smaller co-reference sets (2.9 sets/query in the top10 ranks). Overall, we observe that federation without consolidation performs best when assessing relevance using the *standard* method. We note that however, since the ranked list with consolidation contains sets of entities, it actually captures much more (relevant) results that are not considered when only assessing their labels.

NRel Results. We perform the same analysis as before, but now regard redundant results as not relevant. Different to the std. setting, we observe that the federated system CRM_w performs worse than the best single data source for both movies and the two publication setups. This indicates there were many co-referent results (redundancy) that are not reflected in the results of the std. setting. In summary, when taking redundancy into account, we observe that federation alone no longer improves over the single data sources. If we investigate the combined effect of federation and consolidation with the system MPM, MPM_d, and CRM_c, we observe a different result. Now, the consolidated CRM_c improves

upon the non-consolidated CRM_w system in all cases. Further, CRM_c outperforms both MPM models, which do not always outperform the non-consolidated run. For the 4-sources publ. scenario, we observe that the consolidated run improves upon the non-consolidated runs, but not upon the outlier source MS. In summary, we observed that consolidation helps federated search when redundant results are considered non-relevant.

Expand Results. Also here CRM_c consistently improves upon the MPM models. To see the effect of expanding relevant sets as opposed to only using their labels (and keeping sets with non-relevant labels collapsed), we compare the results of *expand* with the best results obtained in the std. setting, where federated search without consolidation (CRM_w) performed best. We observe that CRM_c also slightly improves upon CRM_w for both scenarios. This means that consolidated federated search (CRM_c) actually outperforms federated search (CRM_w) when the sets' content representing consolidation results are taken into account. Hence, consolidation can even be useful when redundancy is not considered in the evaluation procedure.

Runtime Performance. The main focus of our work is the effectiveness of ranking strategies. We measured the runtime performance of CRM_c on a standard laptop with Intel Core 2 Duo 2.4 GHz CPU, 4 GB memory and 5400rpm HDD. On average, consolidation took 0.7s for an average of 117 entities per query in the movie scenario, and 2.2s for 206 entities in the publication scenario. Ranking took 0.4s for the movie scenario and 1.4s for the publication scenario. These run times are small compared to the amount of time necessary for remote API calls, which took 4s for the publication case and 31s for the movie case. In the latter case, the time includes several API calls because some movie sources return a list of IDs for a query and then each ID has to be fetched individually. This is because these APIs were in fact designed for a different use case (browsing) and are thus, not suitable for online search. In addition, we used developer keys, which may have a lower priority than production keys when requesting data. A demonstrator of our system is available online at `http://km.aifb.kit.edu/services/conesearch`.

6 Related Work

Entity Consolidation is also referred to as record linkage, instance matching or object de-duplication, has a long history in database research and many approaches have been proposed [5–7]. Note, that consolidation is different to fusion, where the goal is to blend instances into one object [21]. With respect to our work, we focus on entity consolidation in a Web context with limited data access. Recent work on consolidating entities for Web search shows that consolidation improves search performance by achieving more diverse and less redundant results [18]. While they use a *supervised* approach relying on training examples, we propose an *unsupervised* approach that is more suitable to uncooperative settings where training data is not available. In a Semantic Web context, the

task of entity consolidation is equivalent to establishing `owl:sameAs` links between entities. Using statistics derived from the entire datasets to establish such links as been studied by [22], who present a self-learning approach, by [23], who focus on scalability, and by [24], who propose statistics as well as logic-based approaches to match entities. However, all these unsupervised approaches are not targeting at search, but have the goal to integrate entire datasets. Hence, they require access to the full datasets, while our approach only uses data retrieved for a given search query. The aspect of *query-time* data integration has been studied for schema alignment during Web data search [11], to identify and consolidate records helpful to process a database query [25], and in the context of probabilistic databases [26].

Entity Search has been studied in many approaches [1, 27–30]. The aforementioned approaches assume a central index comprising the entire data collection. Integrating the data of several sources has been studied in vertical search [31], where the results of different verticals are combined at the front-end level but not at the level of the search algorithms.

Federated Search (distributed IR) has been thoroughly studied for document retrieval [8, 9, 32], where the unit of retrieval are textual documents and not structured entities. *Query translation* for federated entity search has been investigated by [30], who use source-specific query generators to adapt a structured query to each source. *Source selection* and ranking algorithms are studied in [27]. We use the *rank aggregation* strategy of [12] as baseline in our experiment. It requires the presents of coreferences to form a consensus ranking. Our work differs by consolidating entities at query-time and incorporating content, structure and the original rank into the ranking strategy.

7 Conclusion

We have presented the first unsupervised solution for federated entity search using on-the-fly consolidation for uncooperative environments. Our consolidation as well as our ranking technique are incorporated into the language model based IR framework and operate without prior knowledge or training examples, but only on the data obtained for one query and hence are suitable for search over Web data sources with access through APIs. Our experiments investigate the effects of consolidation and federation on retrieval performance. The results show that our approach outperforms a state-of-the-art preference aggregation strategy and that consolidation improves the retrieval performance.

Acknowledgements. This work was partially supported by the EU FP7 project XLIKE (Grant 288342).

References

1. Balog, K., Carmel, D., de Vries, A.P., Herzig, D.M., Mika, P., Roitman, H., Schenkel, R., Serdyukov, P., Tran Duc, T. (eds.): Proc. 1st Int. Workshop on Entity-Oriented and Semantic Search. JIWES, SIGIR (2012)
2. Guo, J., Xu, G., Cheng, X., Li, H.: Named entity recognition in query. In: SIGIR, pp. 267–274 (2009)
3. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: WWW, pp. 771–780 (2010)
4. Wick, M.L., Singh, S., McCallum, A.: A discriminative hierarchical model for fast coreference at large scale. ACL (1), 379–388 (2012)
5. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Magazine 26(1), 83–94 (2005)
6. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Trans. Knowl. Data Eng. 19(1), 1–16 (2007)
7. Köpcke, H., Rahm, E.: Frameworks for entity matching: A comparison. Data Knowl. Eng. 69(2), 197–210 (2010)
8. Callan, J.: Distributed information retrieval. In: Croft, W. (ed.) Advances in Information Retrieval. The Inf. Retrieval Series, vol. 7, pp. 127–150. Springer (2000)
9. Shokouhi, M., Si, L.: Federated search. Foundations and Trends in Information Retrieval 5(1), 1–102 (2011)
10. Lavrenko, V.: A generative theory of relevance. Springer, Berlin (2009)
11. Herzig, D.M., Tran, T.: Heterogeneous web data search using relevance-based on the fly data integration. In: WWW, pp. 141–150 (2012)
12. Volkovs, M., Zemel, R.S.: A flexible generative model for preference aggregation. In: WWW, pp. 479–488 (2012)
13. Chaudhuri, S., Chen, B.C., Ganti, V., Kaushik, R.: Example-driven design of efficient record matching queries. In: VLDB, pp. 327–338 (2007)
14. Neumayer, R., Balog, K., Nørvåg, K.: On the modeling of entities for ad-hoc entity search in the web of data. In: Baeza-Yates, R., de Vries, A.P., Zaragoza, H., Cambazoglu, B.B., Murdock, V., Lempel, R., Silvestri, F. (eds.) ECIR 2012. LNCS, vol. 7224, pp. 133–145. Springer, Heidelberg (2012)
15. Blanco, R., Halpin, H., Herzig, D.M., Mika, P., Pound, J., Thompson, H.S., Tran, D.T.: Repeatable and reliable search system evaluation using crowdsourcing. In: SIGIR, pp. 923–932 (2011)
16. Hayes, A.F., Krippendorff, K.: Answering the call for a standard reliability measure for coding data. Communication Methods and Measures 1(1), 77–89 (2007)
17. Krippendorff, K.: Reliability in content analysis. Human Communication Research 30(3), 411–433 (2004)
18. Dalton, J., Blanco, R., Mika, P.: Coreference aware web object retrieval. In: CIKM, pp. 211–220 (2011)
19. Smucker, M.D., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: CIKM, pp. 623–632 (2007)
20. Leuski, A.: Evaluating document clustering for interactive information retrieval. In: CIKM, pp. 33–40 (2001)
21. Rahm, E., Thor, A., Aumueller, D., Do, H.H., Golovin, N., Kirsten, T.: ifuice - information fusion utilizing instance correspondences and peer mappings. In: WebDB, pp. 7–12 (2005)
22. Hu, W., Chen, J., Qu, Y.: A self-training approach for resolving object coreference on the semantic web. In: WWW, pp. 87–96 (2011)

23. Song, D., Heflin, J.: Automatically generating data linkages using a domain-independent candidate selection approach. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 649–664. Springer, Heidelberg (2011)
24. Hogan, A., Zimmermann, A., Umbrich, J., Polleres, A., Decker, S.: Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. J. Web Sem. 10, 76–110 (2012)
25. Bhattacharya, I., Getoor, L.: Query-time entity resolution. J. Artif. Intell. Res. (JAIR) 30, 621–657 (2007)
26. Ioannou, E., Nejdl, W., Niederée, C., Velegrakis, Y.: On-the-fly entity-aware query processing in the presence of linkage. PVLDB 3(1), 429–438 (2010)
27. Balog, K., Neumayer, R., Nørvåg, K.: Collection ranking and selection for federated entity search. In: Calderón-Benavides, L., González-Caro, C., Chávez, E., Ziviani, N. (eds.) SPIRE 2012. LNCS, vol. 7608, pp. 73–85. Springer, Heidelberg (2012)
28. Blanco, R., Mika, P., Vigna, S.: Effective and efficient entity search in rdf data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 83–97. Springer, Heidelberg (2011)
29. Cheng, T., Yan, X., Chang, K.C.C.: Entityrank: Searching entities directly and holistically. In: VLDB, pp. 387–398 (2007)
30. Endrullis, S., Thor, A., Rahm, E.: Entity search strategies for mashup applications. In: ICDE, pp. 66–77 (2012)
31. Arguello, J., Diaz, F., Callan, J.: Learning to aggregate vertical results into web search results. In: CIKM, pp. 201–210 (2011)
32. Nguyen, D., Demeester, T., Trieschnigg, D., Hiemstra, D.: Federated search in the wild: the combined power of over a hundred search engines. In: CIKM, pp. 1874–1878 (2012)