Coreference Aware Web Object Retrieval

Jeffrey Dalton*
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
Amherst, Massachusetts
jdalton@cs.umass.edu

Peter Mika and Roi Blanco
Yahoo Research
Barcelona, Spain
{pmika, roi}@yahoo-inc.com

ABSTRACT

As user demands become increasingly sophisticated, search engines today are competing in more than just returning document results from the Web. One area of competition is providing web object results from structured data extracted from a multitude of information sources. We address the problem of performing keyword retrieval over a collection of objects containing a large degree of duplication as different Web-based information sources provide descriptions of the same object. We develop a method for coreference aware retrieval that performs topic-specific coreference resolution on retrieved objects in order to improve object search results. Our results demonstrate that coreference has a significant impact on the effectiveness of retrieval in the domain of local search. Our results show that a coreference aware system outperforms naive object retrieval by more than 20% in P5 and P10.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Selection Process

General Terms: Algorithms, Experimentation

Keywords: Semantic Search, Vertical Search, Structured Data, Object Retrieval, Coreference

1. INTRODUCTION

Web search engines compete today by looking for ways to provide specific results to their users beyond the familiar list of "ten blue links." One approach pursued in the past is to integrate object search results from vertical search databases maintained by the provider of the Web search engine. An example is Yahoo! Local, a vertical for business listings, which searches over a curated collection of structured data sourced from multiple trusted providers. Though costly, careful aggregation of the underlying data feeds also ensures that there are no duplicate business listings in the data set. Results from Yahoo! Local are integrated into the main search engine through an information box that appears on top of the search results, and shows a small number of related results from vertical search.

A significant problem with this approach is that vertical search engines built in this manner suffer from a lack of coverage compared to Web search and rely on expensive data feeds sourced from commercial providers. In this context, the emerging Web of Data provides an opportunity for search providers to significantly extend the coverage of their vertical search results, and also to create more compelling search result presentations [13]. Based on the success of the Semantic Web and particular efforts such as microformats[1] and Linked Data[2], the Web of Data has grown considerably in size in the past years [14]. In addition, improved methods of Information Extraction allow web-scale extraction of information with increasing accuracy [6].

In this work, we address a crucial challenge of this approach: the presence of multiple descriptions of the same object from multiple sources. The problem is demonstrated on the case of Yahoo!, where the search for business listings extracted from the Web of Data is offered as an option on the left bar of the search engine, shown in Figure 1. The retrieval base of this search is significantly larger than that of the curated Yahoo! Local database. It searches over structured business objects extracted from the web. A typical query such as *pizza Amherst, MA* returns 926 results in Yahoo! Local, while Yahoo!'s web object search returns 6,420 local business objects. However, as Figure 1 also shows, greater recall comes at the price of *coreferent* results, i.e. object results that describe the same real-world entity. Figure 1 shows that four of the top results are for Antonio's pizza and that Athena's Pizza occurs twice. There are only six unique objects on the first page of search results. This search experience could be significantly improved by performing coreference resolution on the objects in the search collection, either globally or in response to the query.

To address these shortcomings, we introduce the concept of *coreference aware object retrieval*. The context of our work is Ad-hoc Object Retrieval (AOR), which is different from text retrieval in a number of crucial aspects. First, the objects consist mainly of structured attributes and links to other entities, with only a few short pieces of text. Crucially, the coreference of two objects is determined not only by textual features but also by key structural differences. The nature of the search tasks over these objects is also different because the objects contain structured and actionable values: location, phone number, rating, and price informa-

*Work performed while intern at Yahoo! Research.

[1]http://microformats.org
[2]http://linkeddata.org

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

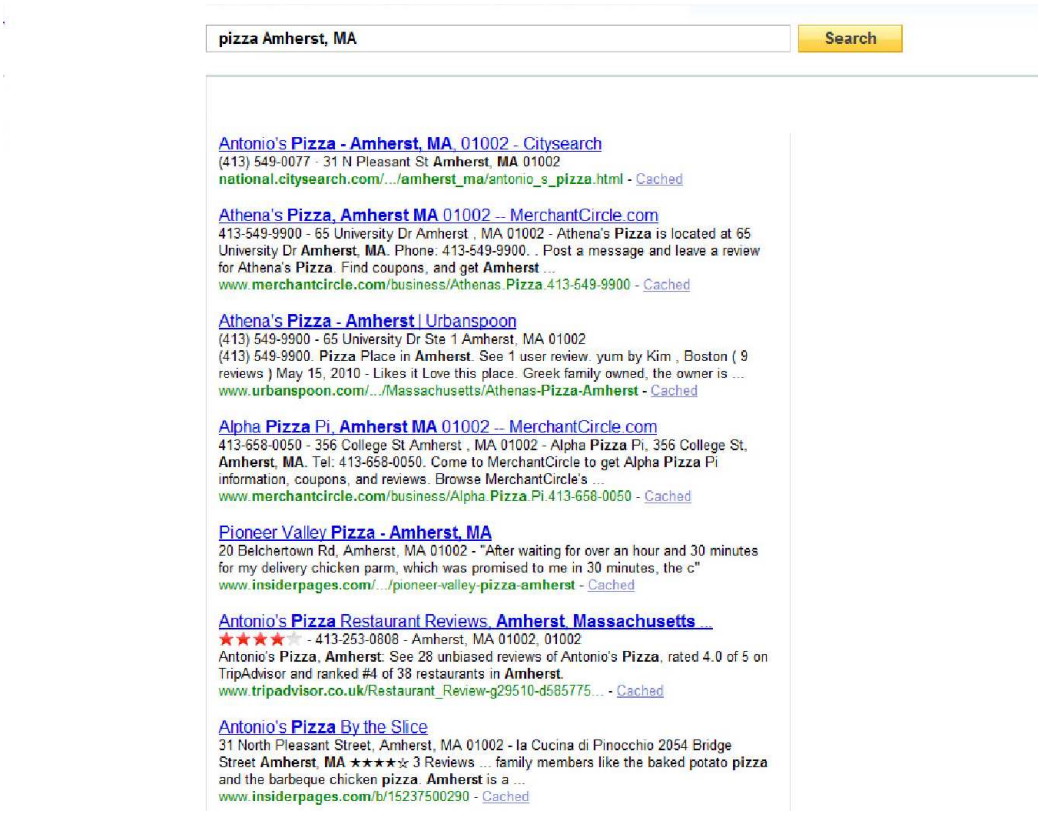

Figure 1: Local business listing objects for *pizza Amherst MA*.

tion. We explain these differences in more detail in Section 2 where we discuss AOR.

The key contribution of our work is to establish coreference aware information retrieval and show the following:

- A substantial amount of redundancy exists in existing object data that can be retrieved from the Web of Data, indicating a large potential impact on applications using the data.
- Demonstrating that coreference aware evaluation has a significant impact on retrieval effectiveness. In particular, systems that do not handle coreference, dramatically overestimate their effectiveness when coreference is considered.
- Coreference aware retrieval approaches outperform naive IR systems by increasing the unique objects returned in the search results. The results also show that errors made by state-of-the-art coreference systems have a significant effect on retrieval and motivates further work considering the tasks jointly.

The area of our study is local search, i.e. search queries where the user is explicitly looking for local businesses and services. However, our conclusions apply to all web search engines that exploit structured data from the Web in any domain of retrieval.

2. PRELIMINARIES

2.1 Ad-hoc Object Retrieval

Pound *et al.* [19] are the first to provide a formal treatment of object retrieval from an information retrieval perspective. They formalize the problem as the Ad-hoc Object Retrieval (AOR) task in Web of Data collections. They define the AOR task as follows:

- INPUT: a user query (a keyword query, without structure) q which has query type t and query intent z, and a data graph G.
- OUTPUT: a ranked list of object identifiers $o = (o_1; o_2; ...o_k)$ such that each o_i occurs in G.
- EVALUATION: each object o_i is labeled with a score (independently of the rest) by a judge with access to all the information contained in or linked to by o_i, with respect to the query q, query type t, and the query intent z.

Based on an analysis of query logs, they suggest a typology of web queries and note that the two most prevalent types of queries are **Entity** and **Type** queries. In an Entity query, the intention of the user is to find references to a particular real-world entity. In a Type query, the intention is to find entities of a particular type or class. The process of evaluating an object retrieval system is similar to web search evaluation. The objects are rated on a four point scale for relevance to the query (Perfect, Good, Fair, Not Relevant).

Field	tok1	tok2	tok3	tok4
vcard:fn	Antonio's	Pizza		
vcard:street-address	31	N	Pleasant	St
vcard:locality	Amherst			
vcard:region	MA			
vcard:postal-code	1002			
vcard:tel	413	253	808	

Table 1: Example of tokenized hCard data mapped to indexable fields

2.2 Ranking models for AOR

In order to test the impact of coreference on retrieval we use well-known retrieval models proven to be effective. Specifically, models based on the Markov Random Field retrieval model (MRF-IR) [12] using unigram and sequential dependence models. We also use the probabilistic BM25 model. In the recent SemSearch 2010 Workshop these methods were the basis for the most effective object retrieval systems for the Entity Search track [10].

The AOR task models typical search workloads performed by semantic search engines, which provide the ability to retrieve resources based on words that appear in the values of properties. The data collection used in retrieval is represented using the Resource Description Framework (RDF) to model it as a series of triples (resource, property, value), also known as (subject, predicate, object). The values may contain textual content or refer to the URI of other resources. To construct objects, triples are grouped by resource. Each property (also referred to as a predicate) is mapped onto fields which can then be tokenized where appropriate to support keyword matching. An example of an object derived from hCard microformat data is presented in Table 1.

We now describe the retrieval models utilized to rank a structured object O, contained in a collection C of $|N|$ objects, with respect to a keyword query Q formed of $q_1, \dots q_n$ query terms. The models we use do not leverage the RDF property structure for field matching or weighting. The selected models can be applied across all Web of Data objects and represent typical baseline retrieval systems. The effectiveness of more advanced models is inconsistent across collections and is very sensitive to parameter tuning.

2.2.1 BM25

BM25 is a probabilistic retrieval model that is an effective approximation of the two-poisson model of term frequency distributions. For an object O the score with respect to a query Q is defined as:

$$s(Q,O) = \sum_{i=1}^{|Q|} \log \frac{N - tf(q_i,C) + 0.5}{tf(q_i,C) + 0.5} \cdot \tag{1}$$

$$\cdot \frac{(k_1+1)\cdot tf(q_i,O)}{k_1 \cdot (1 - b + b \cdot \frac{N}{dl(O)}) + tf(q_i,O)} , \tag{2}$$

where $dl(O)$ is the length of the object in tokens, $tf(q_i,O)$ the number of times q_i occurs in O, $tf(q_i,C)$ the number of times q_i occurs in C and b and k_1 are parameters. We set k_1 to its default value of 1.2 as it has little effect in retrieval performance.

2.2.2 MRF-IR

Two standard retrieval models in the MFR-IR framework are the unigram and sequential dependence models. The unigram model treats the document as a bag-of-words with the underlying assumption of term independence. The probability with Dirichlet smoothing is rank-equivalent to the following score:

$$\log p(Q|O) = \sum_{i=1}^{|Q|} \log \frac{tf(q_i,O) + \mu \frac{c_{q,i}}{|C|}}{|O| + \mu} \tag{3}$$

where $c_{q,i}$ is the number of times a word occurs in a collection of documents, $|C|$ is the number of words in the collection, and μ is the smoothing parameter that is set empirically.

The sequential dependence model relaxes the term independence assumption allowing for adjacent term dependencies. It uses the following formulation:

$$s(Q|O) = \lambda_T \textstyle\sum_{i=1}^{|Q|} f_T(q_i,O) + \tag{4}$$

$$+\lambda_{OW} \textstyle\sum_{q_i,q_{i+1}\in Q} f_{OW}(q_i,q_{i+1},O) + \tag{5}$$

$$+\lambda_{UW} \textstyle\sum_{q_i,q_{i+1}\in Q} f_U(q_i,q_{i+1},O) , \tag{6}$$

where

$$f_T(q_i,O) = \log\left[\frac{tf(q_i,O) + \mu \cdot \frac{tf(q_i,C)}{N}}{dl(O) + \mu}\right] \tag{7}$$

and f_{OW} and f_{UW} are computed like f_T but replacing the $tf(q_i,\cdot)$ function with the count $tf(q_i,q_{i+1},\cdot)$ where q_i and q_{i+1} appear ordered and unordered windows in the text respectively. λ_T, λ_{OW}, λ_{UW} are weighting parameters as suggested by Metzler *et al.* [12].

3. RELATED WORK

We now consider work related to keyword search over structured objects. We also discuss the relationship to previous work on redundancy and diversity in information retrieval. To our knowledge, ours is the first work to address the impact of coreference in object retrieval.

3.1 Keyword search over structured data

The area of keyword search over structured objects is well studied. At a recent SIGMOD conference, Chen [7] provides an overview of recent work in the database community. Agrawal *et al* [2] and Paprizos *et al* [16] both address the problem of returning structured objects in response to web keyword queries. These works assume a clean database where coreference, if necessary, has been performed and therefore do not address coreference.

3.2 Diversity in retrieval

There is significant recent focus on the problems of diversity and redundancy in document retrieval. The work in this area deals primarily with unstructured and semi-structured documents without clearly defined structured relationships. Our work differs from previous studies because we focus on objects in the Web of Data where the relationships between objects is well-defined. The subject of diversity received attention at a recent SIGIR workshop [20] on the topic. The workshop identified two classes of diversity, *extrinsic diversity* and *intrinsic diversity*. Extrinsic diversity models uncertainty about the information need due to query ambiguity

(e.g. pumps). Intrinsic diversity focuses on avoiding redundancy by presenting novel and useful results to a well defined need. The problem of coreferent objects is specific form of intrinsic diversity for structured object retrieval.

Recently developed are "diversity aware" retrieval models that combine relevance scores with inter-document similarity for retrieval over unstructured documents. An important early work is that of Carbonell and Goldstein [5] who define Maximal Marginal Relevance (MMR). MMR combines relevance and novelty by ranking documents according to both their similarity to the query and their dissimilarity to other documents. They performed a small scale evaluation on 42 queries and used content based features to define similarity. We apply a similar approach to retrieval over structured objects and explicitly model the object coreference relationship. Clarke *et al.* [8] study diversification of results for Question Answering. They develop α-nDCG which models documents containing information nuggets and relevance as a function of those nuggets. The issue of coreference does not conceptually fit well with the 'nugget' based approach for unstructured documents.

Bernstein and Zobel measure the impact of syntactically redundant documents on the TREC GOV1 and GOV2 collections [3]. As our results show, although some coreferent documents are duplicates, there is a significant amount of non-trivial object redundancy in Web of Data collections. For web search, a number of approaches have recently been developed to address minimizing risk by diversifying search results [26] [24] [21]. These studies focus on exploiting topical and host-based correlation between documents. In our work we instead focus on the structured coreference relationships between documents.

There have also been attempts to develop evaluation methods for queries requiring extrinsic diversity. These include 'Intent Aware' versions of normalized discounted cumulative gain (NDCG), mean reciprocal rank (MRR), and mean average precision (MAP) [1]. In our study most queries are well defined and do not require extrinsic diversification. We therefore utilize the standard non-diversity aware evaluation measures.

Vee *et al.* [23] demonstrate methods for efficiently computing diverse query results when querying structured data in the context of online shopping. However, they do not evaluate the relevance of the returned objects.

3.3 Object Coreference

Coreference is a common thread across many communities and is referred to as: entity matching, entity disambiguation, cross-document coreference, duplicate record detection, and record linkage. These terms all describe the process for determining whether two records model unique entities. For a survey of various approaches to the problem we refer the interested reader to the recent survey on duplicate detection by Elmagarmid *et al.*[9].

In the database community Benjelloun *et al.* develop a generic framework for merging entities. They demonstrate their algorithm on a collection of 3000 iPod related objects from Yahoo! Shopping and 15,000 hotel records. For efficiency, they perform blocking to limit the number of pairwise comparisons. Nie *et al.* [15] perform entity disambiguation on author objects extracted from the web. In the Semantic Web community Hogan, Harth, and Decker [11] address the problem of object consolidation to merge identifiers of objects across data sources. They perform global object consolidation across a large collection of RDF data where consolidation is performed mostly on Person instances from the FOAF (Friend-of-a-Friend) ontology. There was little overall redundancy. In contrast, for the local business objects extracted from the web we find that there is a significant level of redundancy in returned object results. In addition, all of these works perform global coreference. Our work performs coreference resolution on the scope of retrieved object results in response to a query.

Bhattacharya and Getoor [4] perform query time entity resolution over 'unclean' databases. They describe an 'expand and resolve' strategy for collective resolution. Similarly, we perform object coreference over results returned in response to a ranked keyword query. However, they use structured queries without relevance ranking. Our work differs because we study the impact of coreference on the relevance of returned objects and the interaction between these aspects.

4. COREFERENCE AWARE RETRIEVAL

The problems of object coreference and object retrieval are usually modeled as independent tasks. By considering these tasks together, our goal is to improve the experience for users of object retrieval systems. We now provide an overview of this process.

First, retrieval is first performed over the underlying unclean object collection. Next, coreference classification is performed on the objects retrieved, creating clusters of coreferent objects. Optionally, at this point the clusters may be reranked. Finally, a representative object from each cluster is selected or created by merging objects.

We now examine some of the underlying issues that this process presents. We first discuss underlying indepedence assumptions between objects during retrieval. Next, we explore the impact of coreference error on effectiveness. Finally, we outline additional steps that are necessary beyond the traditional evaluation processes needed to perform coreference aware evaluation for object retrieval.

4.1 Text Retrieval

In conventional text retrieval, a fundamental assumption is that document relevance is independent from other retrieved documents. The probability ranking principle (PRP) states that:

If an IR system's response to each query is a ranking of documents in order of decreasing probability of relevance, the overall effectiveness of the system to its user will be maximized. [22]

However, this is problematic when performing retrieval over *unclean* object data-sets that contain many objects that represent the same real world entity. The effectiveness of a system depends on both the relevance of retrieved objects as well as the number of unique entities presented.

4.2 Object Retrieval

Because of the well-defined coreference relationship between objects, we can model the novelty of a retrieved object in a ranked list. For an object O_k retrieved at position k in a result set we define novelty with respect to the previous retrieved results as follows:

$$novelty(O_k) = \begin{cases} 1 & O_k \text{ is not coref. w/ } O_1, \ldots, O_{k-1} \\ 0 & \text{Otherwise} \end{cases}$$

In other words, only the first retrieved object of a coreference cluster has utility for the user. Subsequent occurrences of coreferent objects do not provide additional information. We must then define a measure that captures this property of an object result set.

We borrow from the data integration community and define the *conciseness* for a retrieved object result set, R. We define the conciseness of R as:

$$conciseness(R) = \frac{NumUniqueObjects \in R}{|R|} \quad (8)$$

An object retrieval system should rank documents in decreasing probability of relevance, $p(Rel|D)$, combined with decreasing probability that the object is coreferent with a previously retrieved document, $p(O_k = unique|O_1, \ldots, O_{k-1})$.

4.3 Coreference Errors

In this section, we examine the impact of coreference error on retrieval effectiveness when the top ranked object is selected as the cluster representative (the other coreferent objects are removed from the final results R).

4.3.1 False Negatives

For a false negative coreference error, an object is classified as non-coreferent when in truth the objects are coreferent. The result of the error is that a redundant object is included in R, lowering the conciseness of the returned results. In this case, whether or not the result was relevant does not matter because coreferent objects of previously retrieved results are redundant and provide no additional benefit.

4.3.2 False Positives

For the false positive case, the object is incorrectly identified as coreferent. The impact on retrieval depends on whether or not the result is relevant. If the object is relevant, then retrieval effectiveness decreases because the relevant object is incorrectly removed from the final result set. In contrast, if the object is non-relevant then removing it does not decrease the retrieval effectiveness. In fact, there is potential for effectiveness to improve because removing a non-relevant object could allow lower-ranked relevant results to be moved higher in the ranked list. Counterintuitively, this means that queries with coreference errors can outperform queries with perfect coreference! The exact impact on effectiveness depends on the number and position of relevant and non-relevant objects incorrectly clustered.

4.4 Utilizing Coreference

Coreference information can also be leveraged during retrieval in other ways. In this section we discuss two possibilities: data fusion and popularity priors.

Data Fusion. In the above formulation all coreferent objects are assumed to be equally reliable and contain identical information. In practice, the objects have different values for their properties and come from sources of varying reliability. In fusion, coreference objects O_c are combined into a single new result in R, O_n. Various data fusion methods have been studied in other previous work [9]. The resulting object can be used to rerank results and possibly display to the users.

Popularity priors. The coreference information can also be used as a feature to indicate object popularity. The correct approach to leveraging coreference depends on the nature of the objects being retrieved, the confidence in coreference information, and the retrieval domain.

For the experiments in Section 5 we do not perform reranking and use the top ranked object as the cluster representative. Exploring the above options is an area for future work.

4.5 Evaluation

To perform coreference aware retrieval evaluation, an additional level of judgments beyond relevance is necessary to determine whether retrieved objects are coreferent with one another. The result is that coreference aware evaluation is significantly more expensive and time consuming than evaluating only topical relevance. In addition to relevance judgments, this evaluation also requires a coreference judgment be made for each result to all previously retrieved objects in R.

In traditional IR evaluation the runs from different systems are pooled and individual result order is ignored. For coreference, the lack of ordering after pooling means that all combinations of documents need to be evaluated. Because the coreference relationship is symmetric, the number of comparisons is defined as the number of unordered pairs of documents: $N \cdot (N-1)/2$. For example, given a query with where $|R| = 10$ the result is 45 coreference judgments are needed.

To reduce the number of judgments needed we utilize several properties. The first is based on transitivity of coreference. If $coref(A, B) = true$ and $coref(B, C) = true$ then this implies $coref(A, C) = true$. We can use this in an online method to reduce the number of positive pairs judged by 1/3. No reduction can be made when the objects are not all coreferent.

A follow-up to this would build on the work of Carterette *et al.* and only compare pairs of objects that could significantly impact the retrieval results enough to change comparative effectiveness of the retrieval algorithms being evaluated.

5. EXPERIMENTS

In this section we describe the results of performing keyword search over a collection of local business objects crawled from the web. We first describe the experimental setup including the query selection process and details on the web object collection. Next, we discuss the evaluation process including relevance assessment and coreference judgments.

We evaluate the object retrieval systems using the Adhoc Object Retrieval (AOR) methodology [19]. Next, we contrast the AOR effectiveness results with an evaluation that is coreference aware. Finally, we show the impact of coreference aware retrieval on these systems.

5.1 Local Object Queries

To evaluate the effectiveness of retrieval we identified a sample of object queries with local intent from the Yahoo! Search query log. Each query was first manually classified as to whether it had local intent. We define queries with local intent as follows:

A query where the goal appears to be an interaction or transaction in a specific geographical location.

Query	Class
morristown west high school	entity
Troy Sports Center	entity
swallows inn	entity
syracuse SPCA	entity
the addison park aberdeen nj	entity
oakland flea market	type
traverse city mi hotel	type
co pug rescue	type
san diego recording studio	type
things to do in toledo ohio	other
map of springfield	other
brewton alabama zip code	attribute

Table 2: Examples of local object queries

AOR class	Count
Entity	659
Type	104
Attribute	2
Relationship	0
Other	11

Table 3: Local queries by AOR class

This definition includes queries which explicitly or implicitly refer to local entities such as business, schools, hospitals, etc. The queries were manually classified by looking at the query keywords and the results returned by a web search engine. We judged results from the *Yahoo! Search Query Log Tiny Sample v1.0* dataset provided as part of the Yahoo! WebScope[3] program. This query set contains 4497 queries sampled randomly from queries submitted at least three times to the Yahoo! US search engine in January, 2009. The log contained 538 queries with implicit and explicit local intent. Because this query set contained fewer local queries than desired, the classification procedure was repeated for another random sample query set containing 7303 queries from Yahoo! Search US query logs from Q3 of 2009. From this set we included 198 queries with explicit local intent.

The local queries were then manually categorized according to the AOR taxonomy described in Section 2.1. The AOR query class breakdown for these queries is shown in Table 3. One finding is that our sample contains a smaller fraction of *Other* and *Relationship* queries than reported from a study of general web query logs [19]. We attribute this to the fact that the local domain is narrower than general web search and more likely to contain references to entities.

From the manually classified local queries we selected a subset to use for testing. Six queries relate to local job intent were filtered out. Random sampling was performed to produce 55 queries from the entity and type query classes for a total of 110 queries. The Type class was oversampled because coreference is particularly important for these queries where users are seeking multiple objects. These are likely to be affected by redundancy in the object collection.

We now describe how the keyword queries were transformed for retrieval. The queries were lowercased and punctuation removed. The queries were spell corrected using the Yahoo! search spelling corrector. A small domain-specific list of stop-words were removed: locator, location, locations, and stores. A gazetteer of state abbreviation names was used to replace full state names to their abbreviations. This is important because local business objects typically contain mailing addresses that utilize the abbreviated form. The queries were not stemmed, but plural terms were depluralized based on the query context [17].

[3]http://webscope.sandbox.yahoo.com/

	Count
Resources (Objects)	106,192,950
Unique Terms	110,564,059
Total terms	1,533,269,855

Table 4: vCard Object Collection Statistics

	Relevant
All	1930
De-duplicated	865
Unique	546

Table 5: Number of relevant objects before and after clustering based on coreference.

5.2 Documents: Local Business Objects

For our experiments, we used local business objects extracted from web pages by Yahoo!'s web search indexing pipeline. The majority of this data is extracted from webpages that use the hCard[4] microformat, an encoding of vCard address book data in HTML pages. A smaller part of the data comes RDFa such as Google's Rich Snippets[5] markup or extracted using proprietary Information Extraction techniques.

The collection was indexed using the Indri[6] search engine. Indri natively supports indexing and retrieval over fielded documents using extents. The data objects were converted to fielded documents and indexed using the procedure described in Section 2.2. No stop-words were removed and stemming was not applied.

As shown in Table 4, the collection contains a sizable 106 million local business objects. The virtual business cards are concise with an average length of 14.4 tokens per object.

5.3 Evaluation Process

The normalized queries were executed using the three retrieval methods described in Section 2.2. The top 20 resources for each query were pooled. This resulted in 4964 unique returned resources to evaluate for relevance. The complete RDF representation of each query-object pair was shown to a human judge and evaluated for relevance. Relevance judgments were made on the four point scale described in Section 2.2. These were converted to binary relevance values where a value of 2 or 3 is considered to be relevant. Six queries were skipped by the human annotator because their intent is unclear. The final query set consists of 104 queries; 54 entity queries and 50 type queries. After relevance assessment, the results were then evaluated for coreference.

To measure coreference, the evaluation procedure described

[4]http://microformats.org/wiki/hcard
[5]http://www.google.com/support/webmasters/bin/answer.py?answer=99170
[6]http://www.lemurproject.org/indri/

	Relevant	MAP	nDCG@15	MRR	P5	P10	P20
BM25	1300	42.55	49.50	59.74	41.73	35.38	29.28
QL	1494	51.19†	55.39†	63.72†	44.81	39.71†	33.37†
SD	1495	54.80†	60.40†	69.83†	50.96†	43.85†	35.38†

Table 7: Object retrieval on all 104 queries. †indicates significance over BM25 at $p < .05$

True Positive	683
True Negative	8381
False Positive	159
False Negative	155
Accuracy	96.65%

Table 6: SVM coreference evaluation

in Section 4.5 was followed. Fully evaluating coreference on the set of pooled objects proved infeasible in practice. There are on average 47.7 unique objects per query and evaluating all pairs would result in over one hundred thousand pairs to annotate. In order to evaluate the difference between coreference aware and traditional evaluation, we fully evaluated all pairs of relevant documents for a query. The presence of redundant non-relevant results is less critical to retrieval effectiveness. Therefore, to reduce annotator effort we did not judge coreference on non-relevant documents.

To evaluate the coreference of the retrieved objects, we performed coreference judgments on the complete relevant set for each query. Across all queries there are 915 unique relevant results, creating 9378 pairs of objects to evaluate. Each pair of objects with their full RDF representation was shown to an annotator who manually evaluated whether or not they were coreferent.

There are 838 pairs of coreferent objects, 8.94% of the total number of pairs. The pairs are used to induce a clustering of coreferent results. The impact of this on the number of relevant objects is shown in Table 5. The number of relevant objects is reduced by 71.7% from all objects and 36.9% from the de-duplicated relevant results. This indicates that there is a significant degree of non-trivial object coreference that could significantly impact the diversity of retrieved results. We now explore the impact of this redundancy on the retrieval process.

5.4 Coreference Classification

In order to identify coreferent objects at retrieval time we trained a support vector machine (SVM) classifier using LIBSVM[7]. The classifier takes as an input a vector that represents the similarity between two objects using a set of individual features, and produces a binary class label that indicates whether the two objects are coreferent or not. Each one of the features in the input vector corresponds to the output of a similarity function over pairs of attributes corresponding to the same property, this is, their *field* similarity. We compute three similarity functions for the attributes in each pair: Levenshtein distance, Jaccard distance over 3-grams and the exact match of numerical fields (such as telephone numbers or zip codes). The final feature vector contains three similarity values for each matching pair of properties. We note that these similarity distances are used in a plethora of applications from database de-duplication and record linkage but also on spell checking and classification. The combination of these metrics with the classifier sufficed to identify coreferent objects.

We mapped the input data $\mathbf{x_i}$ into a higher dimensional space using a radial basis kernel of the form $K(\mathbf{x}, \mathbf{x_i}) = exp(-\gamma||\mathbf{x} - \mathbf{x_i}||^2)$, where γ is a parameter. This kernel is able to handle the case when the relation between class labels and attributes is not linear. SVMs also contain a tunable penalty parameter $C > 0$ that introduces the tolerance of the model upon misclassified instances. Given that both γ and C affect the performance of the SVMs, we learn them from the training data using 10-fold cross validation. Our classification set consists of 9378 pairs of objects which have been manually classified as coreferent or distinct.

The overall results are shown in Table 6, where the accuracy averaged over the 10 folds is higher than 95%. The classifier is able to distinguish coreferent objects using with very high precision and recall. However, in our experiments we will show that the false positive errors are particularly problematic when we remove coreferent objects. Before examining coreference aware retrieval, we first evaluate retrieval using traditional non-coreference aware evaluation.

5.5 Coreference Based Diversification

In this section we measure the object retrieval effectiveness of three widely used object retrieval models: BM25, Query Likelihood (QL), and the Sequential Dependency model (SD) over all objects. The results returned include coreferent pairs of objects. For each method, the top 100 results are retrieved for each of the 104 object queries. We use cross-validation to tune the BM25 b parameter and the μ Dirichlet priors smoothing parameter for QL and SD. We set the SD parameters using the settings in [12], which were shown to be robust to tuning. The b and μ parameters control the influence of document length normalization in the final scores. It is worth noting that the objects we are dealing with are short the parameters have a considerable influence in the final performance of all three methods. Regarding parameter stability, both QL and SD tuned parameters using the folds are close to the over-fitted optimum using the 104 queries ($\mu = 100$). The tuned b for the BM25 runs vary between the folds, although their final performance is less affected by a particular choice of b.

The 2-fold cross-validated results are shown in Table 7. Significance testing is done using the sign test and we report significance at $p < 0.05$. The results show that keyword document retrieval methods perform very well on vCard objects. In particular, the SD model strongly outperforms both the BM25 and QL models, which is consistent with previous findings [12] for document retrieval. The results are promising; however, they do not provide a full and accurate representation of utility for users. The evaluation so far does not consider the impact of redundancy in the search results due to coreferent objects.

[7]http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

	RelRet	MAP	nDCG@15	P5	P10	ΔRelRet	ΔMAP	ΔnDCG@15	ΔP5	ΔP10
BM25	1300	42.55	49.50	41.73	35.38					
BM25-util	340	15.84	28.14	22.69	14.33	-73.85%	-62.77%	-43.15%	-45.63%	-59.50%
QL	1494	51.19	55.39	44.81	39.71					
QL-util	393	18.22	30.61	24.04	15.96	-73.69%	-64.41%	-44.74%	-46.35%	-59.81%
SD	1495	54.80	60.40	50.96	43.85					
SD-util	416	20.57	33.44	26.15	17.50	-72.17%	-62.46%	-44.64%	-48.69%	-60.09%

Table 8: Coreference Aware Evaluation

	MAP	P5	P10	Δmap	ΔP5	ΔP10
BM25	33.82	37.78	33.52			
BM25-util	9.98	22.59	15.37	-70.49%	-40.21%	-54.15%
QL	38.33	38.52	36.85			
QL-util	9.37	23.33	16.67	-75.55%	-39.43%	-54.76%
SD	42.80	47.78	44.44			
SD-util	12.68	26.30	19.63	-70.37%	-44.96%	-55.83%

Table 9: Coreference aware evaluation restricted to the 50 type queries

5.6 Baseline Object Retrieval

We now address the problem of redundancy in object results and demonstrate that it is of critical consideration when evaluating retrieval effectiveness. As discussed in Section 4.2, for this evaluation we assume that coreferent objects provide no additional value over previous objects in the ranked list. Consequently, for coreference aware evaluation only the first occurrence of a unique object is counted as relevant, all subsequent retrieved occurrences are counted as non-relevant. Given that we are measuring the number of *useful* relevant objects, we denote this approach as *util*. The results for this handling of coreferent documents is shown in Table 8, we omit MRR from the figures because the first relevant document is unchanged. The table shows that the number of relevant results retrieved is reduced by more than 70% across all methods. There is a similar effect on all retrieval models independent of their effectiveness. Although the ordering of the systems is unchanged, the differences between the retrieval models is greatly reduced.

Next, we examine the impact of coreference on Type queries specifically (those in which the goal is to find objects of a particular type). The results for only Type queries are shown in Table 9. The baseline retrieval results for the Type queries is lower than the effectiveness of all queries. This indicates that they are more difficult queries. We also observe that the change in MAP is greater than for all queries. The Type queries have on average almost twice the number of relevant objects than for Entity. There is therefore greater opportunity to identify and replace coreferent documents in these queries. The reduction in precision scores is comparable to the overall query set, indicating that the coreferent documents being removed are below result ten.

Overall, the results of coreference aware evaluation show that the impact of coreferent documents on the retrieval effectiveness score is very significant. The effectiveness of the retrieval models id reduced by more than half for the number of returned relevant documents, MAP, and P10. This indicates that there is significant possibility for improvement improvements using coreference aware retrieval.

5.7 Coreference Aware Evaluation

In this section we study the effect of using coreference information to diversify the object results. We first retrieve 100 results per query. To introduce diversity, we test several methods to identify and remove coreferent objects from the results. For diversification we utilize the SVM coreference classifier described in Section 5.4. We also utilize the gold standard annotations of relevant objects.

The result of removing coreferent documents is shown in Table 10. The *ret* column shows the percentage of objects removed by coreference. It shows that between 27% and 33% of the overall number of objects are removed. This represents a substantial reduction in the redundancy of the results. The greater diversity is reflected in improved retrieval effectiveness. We observe that there is a trend for better retrieval models to see greater improvement over the non-diversified results. These models are more likely to replace a coreferent object with one that is relevant. The greatest improvements are the precision scores, P5 and P10. For a web search engine these results are compelling because they represent the first page of objects a user is shown.

Table 10 shows that approximating coreference improves retrieval over the baseline, but is significantly outperformed by using the gold standard judgments on only the relevant documents. In particular, the *relret* column indicates that the SVM is making false positive errors on relevant documents, causing them to be incorrectly removed from the results. The number of incorrectly removed relevant documents is consistent across all retrieval methods. For BM25, 29 objects are removed, for QL 30 objects, and 34 for Sequential Dependency. For a search engine, these mistakes are very costly. However, despite removing some relevant documents, more non-relevant coreferent documents are removed, resulting in improved retrieval effectiveness by moving relevant documents higher in the results.

We now discuss the impact of SVM classifier errors on retrieval in more detail and improve the effectiveness by leveraging the posterior probability estimates. As previously shown, despite the 96.55% accuracy of the SVM coreference classifier, the errors hurt retrieval. In particular, the 159 false positive mistakes result in the incorrect removal of approximately 8% of the relevant results. Therefore, a classifier

	Ret	RelRet	MAP	nDCG@15	P5	P10	ΔRet	ΔRelRet	ΔMAP	ΔnDCG@15	ΔP5	ΔP10
BM25	10400	340	15.84	28.14	22.69	14.33						
BM25-SVM	7562	311	16.72	30.43	26.15	17.69	-27.29%	-8.53%	5.56%	8.14%	15.25%	23.45%
BM25-gt	7591	340	17.48	31.74	27.31	19.23	-27.01%	0.00%	10.35%	12.79%	20.36%	34.19%
QL	10400	393	18.22	30.61	24.04	15.96						
QL-SVM	7386	363	19.38	33.49	29.62	19.81	-28.98%	-7.63%	6.37%	9.41%	23.21%	24.12%
QL-gt	7417	393	20.25	34.66	30.58	21.25	-28.68%	0.00%	11.14%	13.23%	27.20%	33.15%
SD	10400	416	20.57	33.44	26.15	17.50						
SD-SVM	6982	382	22.18	37.39	34.04	22.69	-32.87%	-8.17%	7.83%	11.81%	30.17%	29.66%
SD-gt	7017	416	22.99	38.54	35.00	23.94	-32.53%	0.00%	11.76%	15.25%	33.84%	36.80%

Table 10: Impact of diversifying object results by removing redundant objects. The baselines are coreference aware evaluation without removal. The SVM variants approximates coreference on relevant objects and removes them, the gt removes coreferent objects using the gold standard judgments.

	Ret	Rel Ret	MAP	nDCG@15	P5	P10
BM25-SVM	7562	311	16.72	26.90	26.15	17.69
BM25-SVMpp	7220	364	17.54†	28.06†	26.73	19.23
QL-SVM	7386	363	19.38	30.11	29.62	19.81
QL-SVMpp	6999	431	20.14†	31.13†	30.38	21.92
SD-SVM	6982	382	22.18	32.89	34.04	22.69
SD-SVMpp	6627	448	22.90†	33.93†	35.19	24.33

Table 11: Diversifying object results using an SVM leveraging posterior probability estimates. The SVM variants approximates coreference on relevant objects and removes them, the SVMpp modifies the SVM output using the posterior probability estimates. †indicates significance over SVM at $p < .05$

with fewer false positives could potentially be more effective. We utilize Platt's [18] sigmoid probabilistic outputs as an estimate of posterior probability of the SVM classifier. The errors analysis showed that a significant number of false positive cases were near the SVM threshold. We lowered the classification threshold from .75 to .53. The threshold value of 0.53 was selected based on the distribution of data points to reduce the number of false positive labels. The number of data points below the .53 threshold value increases significantly and a lower value would result in a high number of documents not being labeled correctly as coreferent.

The results comparing the baseline SVM coreference with one leveraging the posterior probabilities is shown in Table 11. The number of false positives is reduced and the number of false negatives is increased. The net effect on retrieval is a small effectiveness improvement across all retrieval methods. The MAP and NDCG improvements are statistically significant.

Performing coreference as a separate step with the SVM creates problems when used in retrieval. The loss function of the SVM does not model the impact of coreference. For example, it does not model position of the objects being considered or the cost of a mistake on retrieval score. Returning a redundant result early in the list is much more costly than returning one later in the list. Considering these two tasks separately is problematic and this study motivates the need to model them jointly. This is analogous to the joint modeling of coreference with other NLP tasks, which has recently shown significant gains [25].

Overall, these experiments demonstrate significant improvement in retrieval effectiveness using coreference to diversify results. The improvements are particularly notable in the precision of the top 5 and top 10 results, which are of particular compelling for keyword retrieval in web search. We show that false positive errors are particularly costly and can be reduced by using the classifier posterior probabilities.

6. CONCLUSIONS

Large Web of Data collections hold the promise to offer significant recall improvements over vertical search engines that leverage only a narrow amount of clean data. In this paper, we considered the problem of performing keyword retrieval over a collection of objects extracted from the Web of Data. We are the first to address the problem of coreference in this setting and to consider the impact in the context of object retrieval. We performed large-scale experiments on a collection of over 100 million real web business objects and local queries from a Web search engine. Our results on a representative sample of state-of-the-art retrieval models show that naive object retrieval systems overestimate their effectiveness by more than 50% when they do not handle coreference. In our experiments, we show retrieval effectiveness improves significantly when redundant objects are removed. The retrieval results improve by 20-40% for precision at 5 and 10. This change reflects a significant improvement in the search experience for users of object retrieval systems.

Coreference aware retrieval approaches significantly outperform traditional IR systems by increasing the diversity of search results. The errors made by state-of-the-art coreference systems have a significant effect on retrieval and motivates further work considering the tasks jointly. In future work, we hope to generalize and verify that these models work across a variety of heterogenous object domains, i.e. that the problem can be addressed in a horizontal fashion. In our experiments we focus on removing redundant coreferent objects. However, other ways of handling coreference, including reranking and data fusion should be considered in the future. Furthermore, relationships other than coreference could be incorporated into our models, and handled

in retrieval such as subsumption of one object result by another. For example, a hotel and related spa and restaurant are distinct but closely related entities.

In document retrieval on the Web, the removal of syntactically identical and very similar webpages is vital. For object retrieval, the same challenge remains for identifying coreference, but with multiple ways of improving the experience for search users.

7. ACKNOWLEDGMENTS

This work is partially supported by the EU Large Scale Integrated Project LivingKnowledge (contract no. 231126).

8. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM '09*, pages 5–14, New York, NY, USA, 2009. ACM.

[2] S. Agrawal, K. Chakrabarti, S. Chaudhuri, V. Ganti, A. C. Konig, and D. Xin. Exploiting web search engines to search structured databases. In *WWW '09*, pages 501–510, New York, NY, USA, 2009. ACM.

[3] Y. Bernstein and J. Zobel. Redundant documents and search effectiveness. In *CIKM '05*, pages 736–743, New York, NY, USA, 2005. ACM.

[4] I. Bhattacharya and L. Getoor. Query-time entity resolution. In *SIGKDD '06*, 2006.

[5] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, pages 335–336, New York, NY, USA, 1998. ACM.

[6] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *WSDM '10*, pages 101–110, New York, NY, USA, 2010. ACM.

[7] Y. Chen, W. Wang, Z. Liu, and X. Lin. Keyword search on structured and semi-structured data. In *SIGMOD '09*, pages 1005–1010, New York, NY, USA, 2009. ACM.

[8] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR '08*, pages 659–666, New York, NY, USA, 2008. ACM.

[9] A. K. Elmagarmid, P. G. Ipeirotis, Vassilios, S. Verykios, A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios, D. Ahmed, K. Elmagarmid, P. G. Ipeirotis, Vassilios, and S. Verykios. Duplicate record detection: A survey. *Transactions on Knowledge and Data Engineering*.

[10] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. S. Thompson, and D. T. Tran. Evaluating Ad-Hoc Object Retrieval. In *Int. Workshop on Evaluation of Semantic Technologies (IWEST 2010), ISWC*, 2010.

[11] A. Hogan. Performing object consolidation on the semantic web data graph. In *In Proceedings of 1st I3: Identity, Identifiers, Identification Workshop*, 2007.

[12] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR '05*, pages 472–479, New York, NY, USA, 2005. ACM.

[13] P. Mika. Anatomy of a searchmonkey. *Nodalities Magazine*, 2008.

[14] P. Mika, E. Meij, and H. Zaragoza. Investigating the semantic gap through query log analysis. In *ISWC '09*, pages 441–455, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] Z. Nie, J.-R. Wen, and W.-Y. Ma. Object-level vertical search. In *CIDR*, pages 235–246, 2007.

[16] S. Paparizos, A. Ntoulas, J. Shafer, and R. Agrawal. Answering web queries using structured data sources. In *SIGMOD '09*, pages 1127–1130, New York, NY, USA, 2009. ACM.

[17] F. Peng, N. Ahmed, X. Li, and Y. Lu. Context sensitive stemming for web search. In *SIGIR '07*, pages 639–646, New York, NY, USA, 2007. ACM.

[18] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[19] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *WWW '10*, pages 771–780, New York, NY, USA, 2010. ACM.

[20] F. Radlinski, P. N. Bennett, B. Carterette, and T. Joachims. Redundancy, diversity and interdependent document relevance. *SIGIR Forum*, 43(2):46–52, 2009.

[21] D. Rafiei, K. Bharat, and A. Shukla. Diversifying web search results. In *WWW '10*, pages 781–790, New York, NY, USA, 2010. ACM.

[22] Robertson, S. E. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

[23] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. In *ICDE '08*, pages 228–236, Washington, DC, USA, 2008. IEEE Computer Society.

[24] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR '09*, pages 115–122, New York, NY, USA, 2009. ACM.

[25] M. L. Wick, K. Rohanimanesh, K. Schultz, and A. McCallum. A unified approach for schema matching, coreference and canonicalization. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 722–730, New York, NY, USA, 2008. ACM.

[26] J. Zhu, J. Wang, I. J. Cox, and M. J. Taylor. Risky business: modeling and exploiting uncertainty in information retrieval. In *SIGIR '09*, pages 99–106, New York, NY, USA, 2009. ACM.