Semantic Web for Search[*]

Jessica Gronski[**]

UC Santa Cruz
jgronski@soe.ucsc.edu

Abstract. Semantic Web data seems like a promising source of information for improving search. While there is some literature about how semantic data should be used to enhance search, there are no positive conclusions about the best approach. This paper surveys existing approaches to semantic web search, describes adapting a TREC benchmark for evaluation, and proposes a learned representation algorithm for using semantic web data in search.

1 Introduction

The Semantic Web(SW)[2] aims to provide a common framework for publishing linked data on the web. While the SW has yet to become a fully adopted technology, the data that is published about web pages, can and should be exploited for improving web search. Improving web search with SW data is interesting not only because search is an important industry but because it provides another compelling reason for wide adoption of the SW.

This paper describes the beginning of my doctoral research on how to use SW data to improve traditional keyword search over documents. The next section describes the existing approaches to search using SW data and concludes with a summary of experimental results from the literature. Section 3 describes the work done to create a benchmark. Section 4 follows with the description of a proposed learned-representation search algorithm. Section 5 concludes with future work for the project.

2 Search Using Semantic Web Data

Regardless of whether the search algorithm retrieves web pages, RDF documents, RDF triples, or linked-data paths, SW search algorithms follow three information retrieval models: Boolean, Vector Space, or Link-based models.

2.1 Boolean Models

In boolean retrieval, documents are modeled as a set of boolean variables indicating whether the document contains a word, a query is modeled as a boolean

[*] All research is funded by Los Alamos National Labs.

[**] Thanks to my advisors Yi Zhang and Herbert Van de Sompel.

A. Bernstein et al. (Eds.): ISWC 2009, LNCS 5823, pp. 957–964, 2009.

proposition, and the retrieval algorithm returns all documents satisfying the boolean proposition. Given the query “cats AND dogs”, a boolean model returns all documents containing both the terms “cats” and “dogs”.

Boolean algorithms will miss documents that use words synonymous to the query terms and also fails to specify the order of documents returned. To address the latter problem, algorithms have ordered documents returned by the number of relevant terms contained [22] or used cover density, the distance between query terms within the document [5].

Many SW search algorithms employ boolean models [17,12,24,21,11] where documents are modeled as sets of variables which correspond to the `WHERE` clauses of a SPARQL query, and the boolean algorithm returns all query-satisfying items. Most do not address the ordering problem, an exception is K-search [3] which orders the boolean retrieved documents using the query’s term frequency in the document text (ignoring the SW data). Others order the boolean retrieved documents using an adaptation of vector-space and link-based models to SW data and are grouped accordingly in subsequent sections.

2.2 Vector Space Models

The vector space model projects document and queries into a vector space with each dimension representing a different term. The ranking function returns documents ordered by measuring their distance to the query in vector space.

Using Euclidean distance for matching the query and document vectors discriminates against long documents so instead normalized cosine distance is used:

$$normcos(\overrightarrow{q}, \overrightarrow{d}) = \frac{\overrightarrow{q} \times \overrightarrow{d}}{||\overrightarrow{q}|| \cdot ||\overrightarrow{d}||}$$

where $\overrightarrow{q}$ and $\overrightarrow{d}$ are the vector representations of the query and document respectively. Alternatives include vector similarity measures which bias for length as it was discovered that in TREC, an established information retrieval benchmark, longer documents were more likely to be relevant[4,25].

The vector space models differ not only in how they compare vectors, but also in their approaches to mapping documents and queries into the term vector space. One approach, TFIDF[4,23], uses term frequency and inverse document frequency, calculates the document’s weight in each of the vector space dimension (recall that each dimension represents a term) as the frequency of that term(t) in the document(d) divided by the frequency of documents in the corpus($\mathit{\Delta}$) containing the term.

$$TFIDF(d,t) = \frac{|t \in d|}{\sum_{\tau} |\tau \in d|} IDF(t)^{-1} \quad \text{where } IDF(t) = \frac{|\{\delta | t \in \delta \wedge \delta \in \mathit{\Delta}\}|}{|\mathit{\Delta}|}$$

The SW search papers which use vector-space models generally incorporate a variation of TFIDF [8] or use frequency directly [1,26] when ranking items. The approach taken by Vallet et. al[8] is to perform a boolean retrieval on the documents by translating the keyword query into a structured SW query. They then order the returned documents using an adaptation of TFIDF where instead of looking at the frequency of terms when calculating TFIDF they look at the frequency of the queried SW class instances. Two systems [1,26] that rank compound direct and indirect paths between SW instances, use the frequency of the constituent relationships to order the paths.

2.3 Link-Based Models

Link-based retrieval models rely on documents having links between them such as web pages or scholarly documents.

The Pagerank algorithm[19] models documents as nodes and HTML links as directed edges between nodes. Pagerank then models the user as a random walker that generally walks along edges and is equally likely to follow each outgoing link. With small probability, the walker will not follow a link but jump to any node with equal probability. Pagerank ranks the documents using the stationary distributions over documents which the random walker creates. Pagerank and its many variants[14,15,10] can be unified as part of the same framework and the variants have been shown to produce scores that are highly correlated[6].

Link-based SW algorithms adapt Pagerank to the SW setting by using the links between SW data [7,9,27,20]. The Swoogle search engine weighs different types of links more than others [7,9]. The SWRank algorithm[27] reverses the SW links when computing Pagerank. Most link-based algorithms use boolean retrieval to narrow the search to query relevant documents [7,9,27], and use their adapted Pagerank scores to order the boolean results. The exception is SWRank which incorporates a term-based TFIDF component to make the scoring query sensitive.

2.4 Experimental Evaluation in Semantic Web Search

Despite the large number of papers on using SW data in search, it is difficult to compare the retrieval performance of each algorithm as all but one of the papers report qualitative, positive results on different datasets and eschew established retrieval benchmarks. Vallet et. al is the notable exception as they used a TREC information retrieval benchmark for evaluation, however they reported negative results for their TFIDF algorithm. This lack of consensus on how to incorporate SW data in search is what motivates this work.

3 Experiment Setup

For evaluating any algorithm, we need an established benchmark ideally containing a set of documents with supporting SW data, a suite of queries, judgments for each document-query pair, and a baseline algorithm capturing the current best approach to the benchmark.

3.1 TREC Blog Track: Documents, Queries and Judgments

As we know of no ideal benchmark we choose to use the BLOG06 dataset created for the TREC benchmark conference's blog retrieval track because though it lacks explicit SW data, the results can be directly compared with competitive information retrieval algorithms.

The BLOG06 dataset was collected during late 2005 and early 2006 and consists of 100,649 feeds, 3,215,171 permalink documents (blog posts), and 324,880 homepages from both top-quality blogs, spam blogs, and manually picked blogs of unknown quality. The TREC blog track has two relevant tasks, the blog and post retrieval tasks (called the distillation and adhoc tasks in the literature). For each task TREC provides a set of questions, a dataset of documents and binary judgments of relevance for a subset of the blog-query or post-query pairs.

We shall evaluate an algorithm's performance using standard TREC metrics mean average precision (MAP), precision at ten documents (P@10), and R-Precision.

3.2 Semantic Web Data Creation

We extract SW data from BLOG06 documents and create SIOC[18] blog ontology classes and links between them which can be summarized on the right side of Figure 1. The two containment relationships `sioc:container_of` and `reply_of` shows that a post is contained in a blog and a comment is a reply of a post respectively. The `sioc:links_to` citation relationship indicates that the origin entity has an HTML link pointing to destination entity (all links, not in a comment or post are assumed in the blog). This data is distinct from ordinary HTML links as they originate from entities rather than web documents and thus the distinction between a informative link in a post and a spam link contained in a comment can be made. As blog-internal `sioc:links_to` data does not convey the same semantic information as links to external sites (often they are to the prior/next entry), we exclude this data.

For the learned graph approach we shall describe this SW data as a multi-relational graph $G = (V, \mathbb{E} = \{E \in V \times V\})$ where the nodes V are the instances of SIOC classes `weblog`, `post`, and `comment` and $\mathbb{E}$ is the set of directed, typed edge matrices E. Each E is a binary adjacency matrix which represents a connection that exists between different entities (either Blogs, posts, or comments) in the dataset. These edge sets are summarized on the right side of Figure 1.

3.3 TREC Baseline

Though the ideal baseline would be the best results reported in the TREC conference, we were unable to parse all blogs and thus cannot use these results. Instead we used the paper of the winners of the TREC 2007 blog distillation task (which used the same BLOG06 dataset) as a guideline to develop an algorithm which approximates their performance on the original data and apply it to our restricted dataset. The resulting baseline performance on both the adhoc

$E \in \mathbb{E}$	Description
$E_{\texttt{w2w}}$	weblog `sioc:links_to` weblogs
$E_{\texttt{p2w}}$	post `sioc:links_to` weblogs
$E_{\texttt{p2p}}$	post `sioc:links_to` posts
$E_{\texttt{c2w}}$	comment `sioc:links_to` weblogs
$E_{\texttt{c2p}}$	comment `sioc:links_to` posts
$E_{\texttt{p}\in\texttt{w}}$	weblog `sioc:container_of` post
$E_{\texttt{c}\in\texttt{p}}$	post `sioc:reply_to` comment

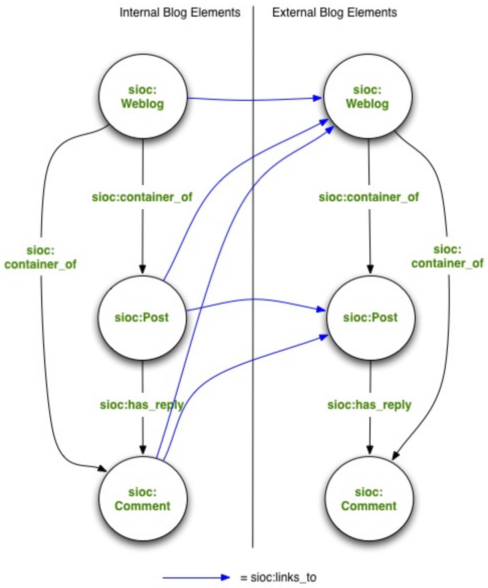

Fig. 1. Links between `(w)eblogs`, `(p)osts`, and `(c)omments` in the TREC dataset

	Data					
	BLOG06 (baseline/reported)			BLOG06 subset (baseline)		
Task	MAP	R-Prec.	P@10	MAP	R-Prec.	P@10
Blog Distillation	0.35/0.34	0.36/0.41	0.42/0.47	0.21	0.28	0.36
Adhoc Task	0.40/0.35	0.41/0.39	0.66/0.56	0.42	0.42	0.62

Fig. 2. On the left the results verify that the recreated TREC 2007 algorithm is close to the conference's reported values. The right column describes the performance of the recreated TREC algorithm on the parsed subset.

and distillation tasks is described in Figure 2 with the left side showing that it gives comparable performance on the entire BLOG06 dataset and the right side showing the baseline's results on the parsed subset.

4 Proposed Learned Representation Approach

We plan on using a learned representation approach for ranking documents with SW data. This link-based approach is based on recent work in the domain of document recommendations by Zhou et. al[28]. Their approach uses a multi-relational graph describing different kinds of links between entities (in their case: scholarly documents, authors and venues) to find latent representations of these entities and then finally produce document recommendations. In our dataset we have a multi-relational graph describing blog entities (blogs, posts, and comments) with HTML link relationships and structural relationships (a post is contained in a blog). Using Zhou's technique and the multi-relational graph, we shall find hidden representations of the blog entities and use the representations as features in a learning to rank algorithm to order the entities.

The approach used to discover the latent representations encodes graph edges as adjacency matrices, with each matrix describing the graph defined by one edge type. The latent representation of the nodes is found by defining a loss function between the adjacency matrices and the hidden representations of the nodes, and choosing the representations which minimize the loss function.

The loss function defined for the citation relationships, such as the post containing an HTML link to another post relationship $E_{\mathtt{p2p}}$, is a laplacian loss function:

$$Loss(X_P) = Tr(X_P^T L(\overline{E_{\mathtt{p2p}}}) X_P)$$

where X_P is the hidden representation of the post nodes and $L(E_{\mathtt{p2p}})$ is the Laplacian of the graph $E_{\mathtt{p2p}}$. The laplacian loss function smooths the difference between adjacent node values and captures the intuition that similar entities will cite one another so the values of their latent representations should be similar.

An undesirable way but effective way to minimize the laplacian loss function above is to use a representation where the values of every node is the same. This makes the function uninteresting and thus the loss function is regularized with $-log|X_P^T X_P|$ to prevent these kinds of simplistic representations.

Besides citation relationships between blogs there also exist containment relationships. For example, the posts in blogs containment graph $E_{\mathtt{p}\in\mathtt{w}}$ is a kind of containment relationship. We define a loss function for containment relationships which captures the intuition that post related by blogs will be close in the latent space:

$$Loss(X_W, X_P) = ||E_{\mathtt{p}\in\mathtt{w}} - X_P X_W^T||_F^2$$

where X_W and X_P are hidden representations of the blog and post entities respectively.

Using these two types of loss functions and taking a linear combination of all loss functions for each edge type we can construct a global loss function to minimize[1]:

$$\begin{aligned} Loss(X_W, X_P, X_C) = {} & k_1 ||E_{\mathtt{p}\in\mathtt{w}} - X_P X_W^T||_F^2 \\ & + k_2 \mathrm{Tr}(X_W^T L(E_{\mathtt{w2w}}) X_W) + k_3 \mathrm{Tr}(X_P^T L(E_{\mathtt{p2p}}) X_P) \\ & + k_4 \mathrm{Tr}(X_W^T L(E_{\mathtt{p}\in\mathtt{w}}^T E_{\mathtt{p2w}}) X_W) - log(|X_W^T X_W| * |X_P^T X_P|) \end{aligned}$$

where the k_i are tunable constants. As the equation is convex the entity representations minimizing the loss function can be found using a nonlinear conjugate gradient(CG) method given the derivatives (see appendix A). The minimal latent representations of the blog entities found will be used as features in a learning to rank algorithm such as Ranking SVM.

[1] For brevity, this loss functions omits relationships with the comment entity. The omitted terms mimic those contained in this representative loss function.

5 Future Work

In order to complete the evaluation of the learned representation approach we need to implement the loss function and apply the macopt[16] convex optimization package to optimize the function. Once our latent representation is found we plan on using Ranking SVM package provided by SVM^{light}[13], to incorporate the latent representation with the baseline algorithm. The algorithm will be considered effective only if the scores produced by ranking SVM will improve on the TREC-based baseline algorithm.

While blog search problem is not as general as web search, the features used by the proposed algorithm (containment and citation links) are not, and we expect will apply to other vertical search over SW data.

Another challenge we hope to address in the future is that the learned representation algorithm in its current form is ontology-sensitive and needs to know what kind of relationship a link-type is (containment or link relationship) to define the loss function. We hope to later develop an ontology-independent loss function or algorithm to deal with a more general environment where the meaning of the relationship is unknown.

References

1. Anyanwu, K., Maduko, A., Sheth, A.: Semrank: ranking complex relationship search results on the semantic web. In: WWW 2005, pp. 117–127. ACM Press, New York (2005)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web: Scientific american. Scientific american (2001)
3. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid search: Effectively combining keywords and semantic searches, pp. 554–568 (2008)
4. Buckley, C., Singhal, A., Mitra, M., Salton, G.: New retrieval approaches using smart: Trec
5. Clarke, C.L.A., Cormack, G.V., Tudhope, E.A.: Relevance ranking for one to three term queries. Inf. Process. Manage. 36(2), 291–311 (2000)
6. Ding, C., He, X., Husbands, P., Zha, H., Simon, H.: Pagerank, HITS and a unified framework for link analysis. Technical Report 49372, LBNL (2002)
7. Ding, L., Finin, T., Joshi, A., Peng, Y., Pan, R., Reddivari, P.: Search on the semantic web. Computer 38(10), 62–69 (2005)
8. Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E., Castells, P.: Semantic search meets the web. In: IEEE Semantic Computing, pp. 253–260 (2008)
9. Finin, T., Mayfield, J., Joshi, A., Cost, R.S., Fink, C.: Information retrieval and the semantic web, p. 113a (2005)
10. Gevrey, J., Ruger, S.M.: Link-based approaches for text retrieval. In: Text REtrieval Conference (2001)
11. Guha, R., Mccool, R., Miller, E.: Semantic search. In: WWW 2003: Proceedings of the 12th international conference on World Wide Web, pp. 700–709. ACM Press, New York (2003)
12. Heflin, J., Hendler, J.: Searching the web with shoe. In: AAAI Workshop 2000, pp. 35–40 (2000)

13. Joachims, T.: Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms. Kluwer Academic Publishers, Norwell (2002)
14. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
15. Lempel, R., Moran, S.: Salsa: the stochastic approach for link-structure analysis. ACM Trans. Inf. Syst. 19(2), 131–160 (2001)
16. Mackay, D.: Macopt, `http://www.inference.phy.cam.ac.uk/mackay/c/macopt.html`
17. Michalowski, M., Ambite, J.L., Thakkar, S., Tuchinda, R., Knoblock, C.A., Minton, S.: Retrieving and semantically integrating heterogeneous data from the web. Intelligent Systems, IEEE 19(3), 72–79 (2004)
18. Möller, K., Bojrs, U., Breslin, J.: Using semantics to enhance the blogging experience, pp. 679–696 (2006)
19. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
20. Patel, C., Supekar, K., Lee, Y., Park, E.K.: Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In: WIDM 2003, pp. 58–61. ACM Press, New York (2003)
21. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: Kim & ndash; a semantic platform for information extraction and retrieval. Nat. Lang. Eng. 10(3-4), 375–392 (2004)
22. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24(5), 513–523 (1988)
23. Salton, G., Fox, E.A., Wu, H.: Extended boolean information retrieval. Commun. ACM 26(11), 1022–1036 (1983)
24. Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y.: Managing semantic content for the web. IEEE Internet Computing 6(4), 80–87 (2002)
25. Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. In: SIGIR 1996, pp. 21–29. ACM, New York (1996)
26. Stojanovic, N., Studer, R., Stojanovic, L.: An approach for the ranking of query results in the semantic web, pp. 500–516 (2003)
27. Wu, G., Li, J.: Swrank: An approach for ranking semantic web reversely and consistently. In: SKG 2007, pp. 116–121. IEEE Computer Society Press, Los Alamitos (2007)
28. Zhou, D., Zhu, S., Yu, K., Song, X., Tseng, B.L., Zha, H., Giles, L.C.: Learning multiple graphs for document recommendations. In: WWW 2008, pp. 141–150. ACM, New York (2008)

A Derivative of Loss Function

$$\frac{\partial Loss}{\partial X_W} = 2(X_P X_W^T - E_{\mathtt{p}\in\mathtt{w}})X_W + 2L(E_{\mathtt{w2w}})X_W + 2L(E_{\mathtt{p}\in\mathtt{w}}^T E_{\mathtt{p2w}})X_W + 2X_W(X_W^T X_W)^{-1}$$
$$\frac{\partial Loss}{\partial X_P} = 2(X_W X_P^T - E_{\mathtt{p}\in\mathtt{w}}^T)X_P + 2L(E_{\mathtt{p}2\mathtt{p}})X_P + 2X_P(X_P^T X_P)^{-1}$$