BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 22 2007, pages 3073–3079
doi:10.1093/bioinformatics/btm452

Data and text mining

Leveraging the structure of the Semantic Web to enhance information retrieval for proteomics

Andrew Smith[2], Kei Cheung[2,3,4,6], Michael Krauthammer[7], Martin Schultz[2] and Mark Gerstein[1,2,5,*]

[1]Department of Molecular Biophysics and Biochemistry, [2]Department of Computer Science, [3]Center for Medical Informatics, [4]Department of Genetics, [5]Program in Computational Biology and Bioinformatics, [6]Department of Anesthesiology and [7]Department of Pathology, Yale University, New Haven, CT, USA

Received on June 17, 2007; revised on July 28, 2007; accepted on August 27, 2007

Advance Access publication October 7, 2007

Associate Editor: Thomas Lengauer

ABSTRACT

Motivation: Proteomics researchers need to be able to quickly retrieve relevant information from the web and the biomedical literature. To improve information retrieval, we leverage the structure of the semantic web, developing an approach for joining it with the largely opposing paradigm of unsupervised web search.

Results: Our approach uses a Resource-Description-Framework (RDF) graph that inter-relates documents through their associated biological identifiers (e.g., protein ID). A search begins with a simple query term (UniProt identifier), which is expanded with terms extracted from documents in the RDF graph surrounding the query (''the subgraph''). We re-rank documents in the full corpus (e.g. all PubMed) by their cosine-similarity scores against a composite word-weight vector created from the subgraph. This vector is a weighted sum of individual word-weight vectors for documents at each node of the subgraph, taking into account the types of relationships between the central query identifier and the nodes connected to it. The computation also uses inverse document frequency (IDF) in a novel way to rescale the local word frequencies in the query's subgraph relative to that in other subgraphs. Applying our procedure to PubMed, we optimize weights for various relationships in the subgraph and benchmark overall performance in detail. Using a subgraph containing family relationships (from PFAM) results in a significant improvement in accuracy (as compared to not considering the subgraph in the search) when assessed against known relationships in the yeast literature. Moreover, we achieve this accuracy using only relatively simple and computationally efficient methods.

Contact: mark.gerstein@yale.edu

Supplementary information: http://hub.gersteinlab.org/ir-supp/

1 INTRODUCTION

1.1 Domain-specific information retrieval for proteomics

Biological research is producing vast amounts of data and information (e.g. from high-throughput experiments such as sequencing projects, microarray experiments and structural genomics) at a prodigious rate. Most of this is made freely available to the public, and this has created a large and growing number of distributed, heterogeneously structured internet and web-accessible biological data and information resources. Some of this is structured or semi-structured, e.g. UniProt (Bairoch *et al.*, 2005), but unstructured data and information, most notably the biomedical literature (PubMed), forms a large and significant source of biological knowledge. Fast, flexible and highly accurate information retrieval from unstructured information sources is an important problem in the life sciences, and in this work we address this in the proteomics context. While general purpose search engines provide basic keyword-based access to such information sources, we believe that much higher accuracy retrieval can be obtained by considering particular domains and leveraging domain-specific knowledge from them. In this work, we extract proteomics domain-specific knowledge from a system we have built called LinkHub (Smith *et al.*, 2007) and use it to perform higher accuracy information retrieval.

Resource Description Framework or RDF (http://www.w3.org/RDF/) is the core technology of the semantic web (Shadbolt *et al.*, 2006) and it models data as a directed labeled graph where the graph's nodes and edges are named by URIs (http://www.w3.org/Addressing/). Our LinkHub system models and stores instance data for a high-level structuring principal or 'scaffold' for biological data, namely biological identifiers (e.g. for proteins, genes, etc.) and the various relationships among them, as a large RDF graph, providing access through web interactive and query interfaces. The nodes of the LinkHub graph represent biological identifiers (e.g. 'P26364', 'GO:0009435', 'PF06052', etc.) and the edges encode relationships among identifiers (e.g. 'protein family member', 'functional annotation', etc.). In addition, a small number of known related web document hyperlinks are attached to the identifier nodes (e.g. for a yeast UniProt protein there would be a hyperlink to its protein entry page at UniProt, its specific page at SGD, etc.), or equivalently these known related documents can be said to be annotated by identifier nodes from the RDF graph.

*To whom correspondence should be addressed.

1.2 Summary

In this work, we address the problem of automated information retrieval of biomedical literature or web documents related to biological identifiers, specifically focusing on UniProt proteomics identifiers for exposition and as a practical use case. The simplest approach for this would be to simply use a search engine (e.g. at PubMed) and do a search using the identifier itself as the search term. However, because of conflated senses of the identifier text, identifier synonyms, and in general a need to consider and query for the key related concepts of the identifier, this will likely not return good results. Note that citations searchable at PubMed rarely contain gene or protein identifiers and PubMed provides no automated retrieval of literature citations relevant to such identifiers (although a small number of citations are manually annotated with such identifiers and can be retrieved by them); Google's Scholar search interface to the scientific literature (http://scholar.google.com) also does not provide effective access as example searches with gene or protein identifiers can demonstrate.

Searching the biomedical literature for biological identifier-related citations using related words and concepts is thus necessary to achieve good results. In this work, we demonstrate how this can be done with high accuracy using related key words extracted from the LinkHub graph. Our general approach is thus to leverage our limited amount of known, semi-structured information about biological identifiers stored in an associated RDF graph (LinkHub) to retrieve relevant documents from the much larger universe of the unstructured biomedical literature or web. The key idea is that the local subgraph radiating out from a given query identifier node (i.e. node corresponding to a UniProt identifier about which it is desired to retrieve additional relevant documents) and the known documents linked to the identifier nodes in that subgraph provide copious information about the query that can be used to improve document retrieval for it. The documents linked to the identifier nodes in the subgraph are considered to be a 'gold standard' training set for what the additional relevant documents should be like, and they are used to construct a function for scoring new documents for how well they match the training set (and hence the query).

The rest of the article is organized as follows. The following section gives the details of our procedure, followed by a section covering the results of an empirical performance assessment of it using a curated yeast bibliography. The article then discusses the results and compares our method with important related works before concluding.

2 METHODS

2.1 Basic procedure for document ranking

Our procedure for document relevance ranking uses basic techniques from information retrieval (Salton and McGill, 1986) and text categorization (Sebastiani, 2002; Williams and Calvo, 2002). We represent documents in the standard vector space model as word weight vectors where the weights are obtained from TF–IDF weighting, and we use the standard cosine similarity metric to measure similarity of documents and/or queries so represented. Term frequency (TF) simply measures the number of occurrences of a word in a document. The inverse document frequency (IDF) weighting factor for a word is log (N/D) where, in the corpus searched, D is the total number of documents and N of them contain that word; the IDF term up-weights infrequent, discriminating words in the corpus and down-weights frequent, less discriminating words.

We first extract the local subgraph surrounding the query identifier from the graph (e.g. 1 level deep in the examples below). We then obtain all the known related documents linked to identifier nodes in this subgraph. We turn each of these documents into word weight vectors and multiply them by their node's weight (which is determined as described below). The pre-IDF step (also described below) can optionally be applied to some or all of these individual word weight vectors. We then form the sum of these individual vectors and call this the combined word weight vector. Finally, the combined word weight vector is re-weighted by a standard IDF step using document frequency statistics for the corpus to be searched (e.g. the PubMed or the web). Some percentage of the lower weighted terms in the combined word weight vector can be optionally eliminated for efficiency. Figure 1 shows a high-level overview of our method. The Supplementary Material gives the formal equations and details for how we construct the combined word weight vector and has examples. To score a new document for relevance, we compute the cosine similarity metric between the word weight vector representation of it and the combined word weight vector.

Finally, it is necessary to perform a retrieval step and obtain documents potentially relevant to the query identifier. A search engine can be used essentially as a keyword-to-document hash [implemented in the internals of the search engine as a so-called inverted index (Witten *et al.*, 1999)] to efficiently obtain such documents. Multiple searches are performed using as search terms all the identifiers for nodes in the subgraph, as well as some number of the top weighted terms from the combined word weight vector; and the top results for each such search are retrieved (e.g. top 50). For the final output, we rank (sort) all the results of these multiple searches together descending based on their computed cosine similarity values against the combined word weight vector.

2.2 Traversing the graph to determine weights

The known documents linked to nodes in the subgraph are not all considered equally important in forming the combined word weight vector, but rather are weighted to reflect their relative importance. In fact, we compute weights for the nodes in the subgraph, and a document's weight is then simply the same as the weight of the node to which it is linked. The central query identifier's node is given the highest importance (weight 1.0), while the other subgraph nodes' weights are scaled down based on their distance from and the types of relationship links connecting them to the query (but synonym links do not incur downscaling). Relationship links (e.g. 'family member' for relationships like 'UniProt → PFAM' or 'functional annotation' for 'UniProt → GO') are specified in LinkHub and we assign numerical weights to them. A node's weight is determined by summing all the weights of nodes linking to it, each multiplied by the weight for the relationship type. This weight calculation for the subgraph starts at the query node (which has weight 1.0) and propagates out to its connected nodes, then their connected nodes, etc. The principled way of setting the relationship weights is through an optimization procedure where we determine the weights that lead to optimal accuracy in retrieving new identifier-related documents. Later we demonstrate this and show how the weights for identifier relationships 'UniProt → PFAM' and 'UniProt → GO' can be empirically optimized.

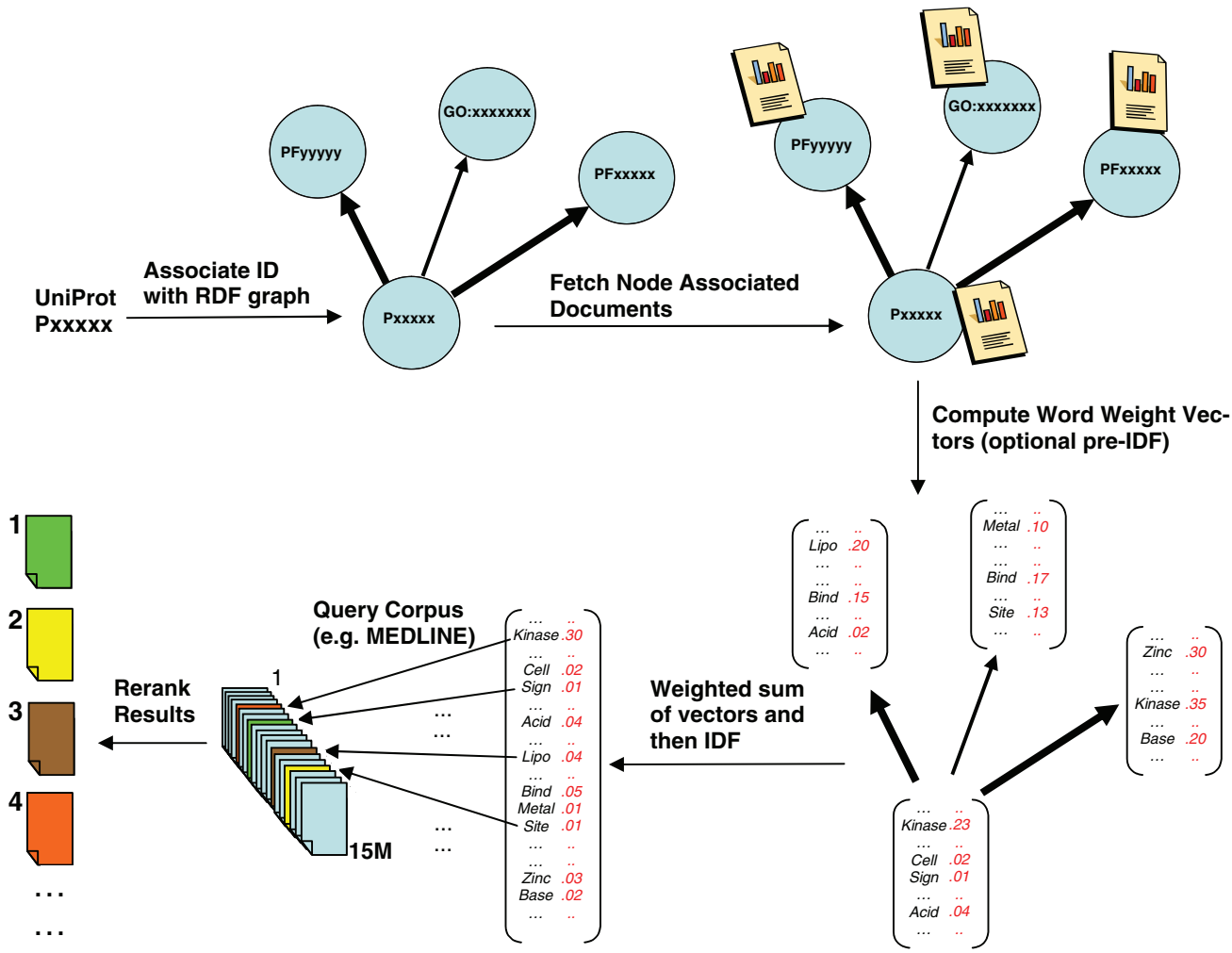

Fig. 1. A high-level overview of our procedure. Note that relationship link thickness is proportional to relationship weight, which we determine through an optimization procedure.

2.3 IDF word weighting based on the graph

We use IDF in a novel way to leverage document-type information from the LinkHub RDF graph for re-weighting individual document's word weight vectors (before summing them to create the combined word weight vector). Documents of the same type (e.g. UniProt protein entry pages) will all use similar general terminology and have the same words from 'template patterns' for the pages. In essence, we want to create word weight vectors for all pages of a type that are maximally different from one another while at the same time each being as specifically relevant and discriminating as possible; we achieve this through IDF. The idea is to do IDF weighting twice, ultimately as usual for the combined word weight vector against the corpus (e.g. PubMed) you are interested in searching but also first for individual documents against document frequencies computed for all (or a random sample of) documents of their same type. We call the first use of IDF the pre-IDF step (because it occurs before the traditional IDF re-weighting against document frequency statistics for the corpus to be searched). Thus, e.g. UniProt (or PFAM, GO, etc.) has many individual, identifier-specific pages which can together be considered a corpus. We compute document frequency statistics for this corpus to use for IDF re-weighting of individual UniProt (or PFAM, GO, etc.) pages. The pre-IDF step has the effect of down-weighting less discriminating words which occur frequently (e.g. *protein* or *sequence* in UniProt pages) while up-weighting words that occur infrequently (e.g. *kinase* in UniProt pages) which are intuitively more discriminating.

2.4 Implementation

We have implemented our procedure in Perl for both the web and the scientific literature (PubMed), although we focus in this article on the literature. To conduct the multiple conventional keyword searches for the scientific literature, we obtained the full PubMed distribution (current up to the end of 2005 and consisting of about 500 XML files of over 15 000 000 total citations) which we indexed and keyword-searched using the open source Swish-e (swish-e.org) application.

Our procedure works well for both web and scientific literature searching, and in the Supplementary Material we give concrete examples (i.e. ranked result lists, example word weight vectors, etc.) of the use of our procedure for the web and PubMed for specific UniProt identifiers and argue for our procedure's better performance compared to the results of conventional techniques (e.g. PubMed's

limited number of manual annotations for them). We also provide the Perl code of our implementation. For PubMed, we provide our code to index and keyword-search it through Swish-e, as well as code to parse returned PubMed XML records. We also provide common code implementing the information retrieval aspects discussed above and for interfacing with the LinkHub system (e.g. LinkHub graph traversal, subgraph extraction, etc.). See the Supplementary Material for details of our procedure's implementation for the web. Finally, we provide the full code for our LinkHub system, as well as a paper and documentation about it.

2.5 Empirical assessment protocol

Here, our important focus is to empirically evaluate our procedure's performance against PubMed which is a crucial unstructured knowledge source for scientists. We use the same evaluation methodology, based on ROCs and AUC statistics, as Aphinyanaphongs *et al.* (2006) work (discussed in more detail below). We base the evaluation on the file gene_literature.tab from the SGD (http://www.yeastgenome.org) FTP site, which provides a large number of relevant citations for yeast proteins. We also form an unrelated group of PubMed citations likely to be completely irrelevant to proteomics by sampling random citations from PubMed which do not appear in gene_literature.tab or as a citation in any UniProt (SwissProt or TrEMBL) entry. For a given yeast protein, its associated citations from gene_literature.tab are called its *in* group, and the citations associated with all other yeast proteins plus the unrelated PubMed citations are its *out* group. We can then reasonably assert that for any given yeast protein the correct relevance ranking of its *in* and *out* groups is: *in*—*out*; to concretely measure the deviation of the word weight vector ranking of the *in* and *out* citations from this assumed correct ranking we use the area under the curve (AUC) of the receiver operating curve (ROC) (Witten and Frank, 2005). The ROC shows us the trade-off in rate of true positive citations (*in* set) versus the rate of false positive citations (*out* set) at each point as the ranked list of citations is scanned from top to bottom. The ROC is beneficial because it can measure performance of a classifier without regard to class distribution or error costs, neither of which we know. The AUC is a single number summary of the ROC with best, maximum value of 1. The quality of search results at the top is very important (e.g. users likely would not look past about the top 100). We thus also separately measure the AUC up to a 5% false positive rate (i.e. the 0.05 AUC with best, maximum value of 0.05) to assess this performance at the top; note that we refer to the normal AUC as the 1.0 AUC to distinguish it from the 0.05 AUC.

2.6 Parameter optimization

We perform experiments separately on random samples of Swiss-Prot and TrEMBL UniProt identifiers. Swiss-Prot is the much smaller part of UniProt, but is of higher quality, having been manually curated. TrEMBL is much larger than Swiss-Prot and is of lower quality due to its being generated by automated processes. For SwissProt and TrEMBL, separately, we pick a random set of identifiers such that each identifier has GO and PFAM annotations and at least 20 citations in gene_literature.tab. We are thus not looking at the full subgraphs for UniProt proteins, but only the subset of the one level deep subgraph containing directly related PFAM and GO identifier nodes (and their associated GO and PFAM identifier-specific pages). As will be shown this was sufficient to obtain very high accuracy, so additional nodes and depth are likely not needed in this case.

We are optimizing the values of four parameters: (1) PW—related PFAM documents' weight; (2) GW—related GO documents' weight; (3) WK—percentage of top weighted words kept in the combined word weight vector and (4) PI—binary value specifying whether pre-IDF is applied or not for all the UniProt, PFAM and GO pages which will together form the combined word weight vectors. We perform a simple grid search optimization procedure at a granularity of 0.1 for all numerical variables being optimized (and 1 or 0 for PI). UniProt entry pages are always weighted 1.0. We might hypothesize that PFAM and GO pages would be more likely to improve information retrieval performance for TrEMBL pages, given that TrEMBL pages have less information since they are not manually curated; also, PFAM and GO pages might not improve (or not significantly improve) performance for SwissProt pages since they are already of high quality, being manually curated, and thus are likely to be fairly complete statements about the proteins they describe. The experimental results, described in the next section, answer these questions.

Finally, we would also like to know if all the tried optimizations lead to statistically significant improvements in performance (as measured by AUC values). In other words even if the optimizations lead to an increase in mean AUC, is this increase significant or likely just due to chance? To assess this we run tests for many UniProt proteins, each multiple times for different combinations of optimizations (i.e. do or not do pre-IDF, different weights for PFAM and GO pages, and percentage of words to keep in the combined word weight vector). We can thus pair the proteins and use the paired Student's *t*-test (Dalgaard, 2002) for statistical significance tests.

3 RESULTS

The Supplementary Material contains a fuller discussion of results and has the complete tables giving the mean .05 and 1.0 AUC values for the randomly sampled TrEMBL and Swiss-Prot proteins, for different trials with different values for the four parameters GW, PW, WK and PI. Table 1 (sorted descending by AUC value) reproduces the important result rows and here we summarize the key results. For both TrEMBL and Swiss-Prot, WK did not seem to make any difference (different values for this made no or negligible change to AUC values) and we thus just show results for the middle value 0.5 of this parameter. Note that this can be taken advantage of for computational efficiency since keeping fewer features requires less computation time of the cosine similarity metric (which increases linearly with the number of features, i.e. length of the word weight vectors). Interestingly, for both TrEMBL and Swiss-Prot, the addition of related GO pages (parameter GW) also did not improve performance (see the Supplementary Material for a discussion of why this may be). Also note that the baseline method against which performance improvements are assessed and stated is where only the UniProt entry page for a query UniProt identifier is used (no related PFAM or GO pages are added) and turned into a word weight vector to which the pre-IDF step is not applied.

3.1 TrEMBL results

The pre-IDF step (parameter PI) gave the largest performance gain, increasing the important 0.05 AUC ~75% and the normal 1.0 AUC 8.4%. In addition, the percentage increases in AUC are greater the farther you go to the left (i.e. as false positive rate decreases) in the ROC; e.g. the 0.01 AUC (not shown) increases over 92%. The AUC increase is thus concentrated in the left portion of the ROC, which is what is desired. The addition of PFAM pages at small weight (parameter PW) gave an additional performance enhancement of ~5% for the 0.05 AUC and 1.2% for the 1.0 AUC. Without the pre-IDF

Table 1. Important mean AUC results for randomly sampled TrEMBL (top) and Swiss-Prot (bottom) proteins

PW	PI	100*AUC
0.05 AUC TrEMBL results		
0.2	1	3.227
0.0	1	3.131
0.6	0	2.053
0.0	0	1.791
1.0 AUC TrEMBL results		
0.1	1	92.742
0.0	1	92.026
0.4	0	86.013
0.0	0	84.931
0.05 AUC SwissProt results		
0.1	1	3.571
0.0	1	3.567
0.2	0	2.525
0.0	0	2.492
1.0 AUC SwissProt results		
0.1	1	95.054
0.0	1	95.025
0.1	0	89.738
0.0	0	89.710

Note that we do not show GW and WK since they did not affect the results (i.e. the optimal value for GW was 0.0 in all cases and changing WK did not appreciably change results). Also, AUC values have been multiplied by 100.

step, the addition of PFAM pages is optimal at larger weight and gives a larger performance increase of almost 15% for 0.05 AUC and 1.3% for 1.0 AUC. The statistical paired *t*-tests (details in the Supplementary Material) all returned highly significant *P*-values, with the largest being 0.003 which is still much smaller than the commonly accepted 0.05 level of significance. Thus, we can conclude that the addition of PFAM pages appropriately weighted and the use of the pre-IDF step both significantly increase information retrieval performance as measured by 0.05 and 1.0 AUC values for UniProt TrEMBL identifiers.

3.2 Swiss-Prot results

The pre-IDF step again gave the largest performance gain, increasing the important 0.05 AUC ~43% and the normal 1.0 AUC 6%. The percentage increases for Swiss-Prot, while still substantial, are not as large as for TrEMBL and this is consistent with the fact that Swiss-Prot, being manually curated, is of higher quality and thus there is less need or room for improvement compared to TrEMBL. However, TrEMBL is much larger than Swiss-Prot and generated by automated processes, and it is thus practically very useful that TrEMBL can be improved more, reaching close to parity with Swiss-Prot. The improvements in both mean 0.05 and 1.0 AUC from the pre-IDF step are also both statistically significant (details in the Supplementary Material). As was conjectured above, the addition of PFAM pages does not help improve performance for Swiss-Prot as much as for TrEMBL. The addition of PFAM pages at small weight very slightly increases mean 0.05 and 1.0 AUC for all cases compared, however, all but one (pre-IDF not applied case) of these increases are not statistically significant (details in the Supplementary Material). Thus, the addition of PFAM pages cannot be said to significantly improve performance.

4 DISCUSSION

4.1 Possible yeast bias

Overall, for a small PFAM page weight and performing the pre-IDF step we achieve 0.05 AUC and 1.0 AUC scores of 0.03227 and 0.9274, respectively for TrEMBL and 0.03571 and 0.9505, respectively for Swiss-Prot, where the maximum possible values are 0.05 and 1.0 for these. Our procedure thus achieves near perfect accuracy and is competitive with state of the art recent methods as we will show below.

Since yeast has been so well studied, using SGD's gene_literature.tab for our empirical evaluation might seem to bias our results. In general, algorithms that learn from data, such as ours, will suffer from 'garbage in, garbage out' and will degrade in performance as training data quantity or quality goes down. However, we have positively shown that given good training data our procedure achieves excellent accuracy. In addition, based on our excellent results for TrEMBL, which is of lower quality than SwissProt and thus likely a good proxy for less well-studied proteins, we can expect our procedure's performance to degrade gracefully with less or lower quality training data.

4.2 General related work

Our procedure can be considered a kind of method for *query expansion* (Mitra *et al*., 1998; Qiu and Frei, 1993; Salton and Buckley, 1990). Here the basic idea is to take an initial query and reformulate it, interactively or by automated means, to improve retrieval performance. Example techniques for doing this include searching also for synonyms of query terms, fixing spelling errors and reweighting original query terms. In our case, we take an initial query (for a UniProt identifier) and greatly expand it, using the background data in LinkHub, into a precise word weight vector containing important keywords related to and descriptive of the identifier. Our work also extends systems like PubMed's 'Related Articles' links (PubMed) to using multiple, weighted documents combined (i.e. from the LinkHub relational subgraph) and demonstrates empirically that this can improve information retrieval accuracy over just using single documents as queries.

In the Supplementary Material, we discuss possible extensions and further uses of our pre-IDF step. For example, we discuss how we could construct a coarse-to-fine cascade of classifiers and consider a pre-IDF step of log relative document frequencies between different levels of the cascade. Interestingly, another recent related work (Suomela and Andrade, 2005) uses a similar but simpler idea, ranking PubMed citations for their similarity to a domain of interest (stem cells in their paper) based on the presence of key words in the title or abstract which are overrepresented in a known

relevant training set (e.g. PubMed citations annotated with stem cell-related MeSH terms) compared to PubMed as a whole.

4.3 Machine-learning classifier approach

A recent related paper to ours showed the high performance of support vector machine (SVM) classifiers trained on gold standard, manually curated bibliographies for specialized information retrieval tasks (Aphinyanaphongs *et al*., 2006) (hereafter referred to as Aphinyanaphongs *et al*.). The Aphinyanaphongs *et al*. work highlighted the need for specialized filters for finding relevant documents in the huge and ever expanding scientific literature. A prominent example of such filters is the manually constructed PubMed Clinical Queries (http://www.ncbi.nlm.nih.gov/entrez/query/static/clinical.shtml). Such a manual approach does not scale well, and in fact Aphinyanaphongs *et al*. demonstrated that superior performance can be achieved automatically by machine-learning SVM classifiers.

In Aphinyanaphongs *et al*. state of the art SVM classifiers were trained on relatively large, manually curated and respected bibliographies of articles in various clinical medicine disciplines, using text from the article title, abstract, journal name and MeSH terms for features. In contrast, the experiments in our work only used the abstract text for features, used only relatively small training sets (i.e. the single UniProt page plus a few GO and PFAM pages), and, in comparison to SVMs, used only a fairly basic classifier model (i.e. word weight vectors compared with the cosine similarity measure). Nevertheless, it is notable that our procedure achieved average AUC scores of 0.9274 and 0.9505 for TrEMBL and Swiss-Prot, respectively, in ranking PubMed documents for their relevance to UniProt proteins which is better than or negligibly smaller than the Aphinyanaphongs *et al*. study which achieved AUC scores of 0.893, 0.932 and 0.966 on three clinical medicine bibliographies. Note that while this is not an exact 'apples to apples' comparison it is still reasonable. We both use the same objective metric, AUC and achieve comparable, near perfect results (thus not leaving much room for improvement) on the same kind of task (ranking PubMed citations), although not on the exact same tasks.

Another noteworthy result of the Aphinyanaphongs *et al*. work is its evaluation of relevance metrics based on citations, such as citation count, journal impact factor and Google's PageRank algorithm. Google's great success was due in large part, at least initially, to its PageRank algorithm (Brin and Page, 1998) which provided an effective solution to the difficult problem of relevance ranking of huge result sets. It would thus seem reasonable to expect that algorithms based on citation information such as PageRank would also prove very effective for information retrieval of the scientific literature, but the Aphinyanaphongs *et al*. work finds this to not be the case, finding their SVM classifiers superior to all citation-based metrics and that adding citation metrics as features only marginally or not at all improved performance. In fact, the Aphinyanaphongs *et al*. work did not directly compare to PageRank, but they cite a previous study (Bernstam *et al*., 2006) that showed citation count superior to PageRank, and since their study directly showed machine-learning classifiers to outperform citation count they conclude by transitivity that machine-learning classifiers are very likely superior to PageRank. Because of favorable performance on the same kind of task compared to the Aphinyanaphongs *et al*.'s work, we can indirectly compare our method to relevance metrics based on citations and infer that our procedure likely would outperform them.

4.4 The Semantic web and search engines: structured versus unstructured search

Search engines and the semantic web can be viewed as two opposing paradigms for information retrieval on the internet, with search engines allowing maximal flexibility of information expression (free text HTML pages) but providing low precision of retrieval (albeit vast, close to complete web coverage). The semantic web requires more rigidity of data expression by prescribing web data to be expressed in fine-grained structured ways but has the benefit of supporting very precise, cross-resource information requests. The drawback of the semantic web is that, to achieve such fine-grained information modeling, people must change the way they create their content to conform to very precise structures, and this is a hindrance to widespread dissemination of semantic web content.

Our work here can be viewed as taking a proactive approach by trying to leverage available semantic web data to enhance knowledge acquisition from and information requests against unstructured sources such as the biomedical literature and the standard web, i.e. information retrieval or web search. We have previously sketched out this basic idea (Smith and Gerstein 2006) and here we attempted to concretely implement it.

The high-level idea is that the semantic web provides detailed information about standardized terms and their interrelationships, and, importantly, unstructured documents can be annotated with those terms as metadata. The terms, their relationships, and the documents that they annotate provide copious information to perform precise information retrieval or web search for free-text documents relevant to those terms (and related terms). We explored this idea here concretely in the proteomics context. Since searching is widely perceived to be a crucial web application, the semantic web's potential to improve it could be of high practical value and an important driving force to help more fully realize the vision of the semantic web.

4.5 Computational complexity

There are important differences of our work compared to Aphinyanaphongs *et al*. First, while we both take a machine-learning filter approach our work demonstrates how specialized filters can be constructed automatically and easily at very large scale (i.e. for the millions of proteomics identifiers present in the RDF graph) using only a relatively small amount of information (i.e. the small number of known documents linked to relational subgraphs' identifier nodes versus a relatively large number of documents in manually created medical

bibliographies). In addition, our method uses relatively simple and computationally efficient methods: IDF plus combined word weight vectors, which to create will have linear time complexity in the number of words. In contrast, Aphinyanaphongs *et al*. use state of the art SVM classifiers which are considerably more computationally intensive and require the solution of a constrained convex quadratic programming optimization problem which has quadratic complexity in memory usage and cubic time complexity in number of training examples (Tsang *et al*., 2005). In spite of this, our method achieves very high accuracy, on par with or better than the SVM-based methods. Users expect text search to be fast and interactive, and thus the simplicity and computational efficiency of our method, and the relatively small amount of information needed for its training, while still achieving competitive, high accuracy is important. Despite these noted differences, the Aphinyanaphongs *et al*. work is consistent with and supports the general approach taken in our work of creating specialized machine-learning filters (in the form of word weight vectors in our case) for retrieval of documents specific to particular proteomics identifiers, and demonstrates the approach's effectiveness.

5 CONCLUSION

In this work, we addressed the important problem of information retrieval for proteomics-related documents, particularly related to UniProt identifiers, and demonstrated several ways to leverage domain-specific data in an RDF graph of biological identifier relationships for this. We empirically demonstrated our procedure's high accuracy against PubMed using a curated bibliography of yeast protein-specific citations. Our procedure's accuracy compares favorably with similar recent related work but is advantageous in using only relatively simple and computationally efficient methods, and a relatively small amount of information that can be leveraged automatically and at large-scale from the RDF graph.

ACKNOWLEDGEMENT

Supported by NIH/NIGMS grant 1U54GM074958-01.

Conflict of Interest: none declared.

REFERENCES

Aphinyanaphongs,Y. *et al*. (2006) A comparison of citation metrics to machine learning filters for the identification of high quality MEDLINE documents. *J. Am. Med. Inform. Assoc*., **13**, 446–455.

Bairoch,A. *et al*. (2005) The Universal Protein Resource (UniProt). *Nucleic Acids Res*., **33**, 154–159.

Bernstam,E.V. *et al*. (2006) Using citation data to improve retrieval from MEDLINE. *J. Am. Med. Inform. Assoc*., **13**, 96–105.

Brin,S. and Page,L. (1998) The anatomy of a large-scale hypertextual web search engine. *WWW7/Comput. Networks*, **30**, 107–117.

Dalgaard,P. (2002) *Introductory Statistics with R*. Springer, New York.

Mitra,M. *et al*. (1998) Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 206–214.

PubMed Computation of Related Articles. http://www.ncbi.nlm.nih.gov/entrez/query/static/computation.html

Qiu,Y. and Frei,H. (1993) Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 160–169.

Salton,G. and Buckley,C. (1990) Improving retrieval performance by relevance feedback. *J. Am. Soc.Inf. Sci*., **41**, 288–297.

Salton,G. and McGill,M. (1986) *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. New York, USA.

Sebastiani,F. (2002) Machine learning in automated text categorization. *ACM Comput. Surv. (CSUR)*, **34**, 1–47.

Shadbolt,N. *et al*. (2006) The semantic web revisited. *IEEE Intell. Syst*., **21**, 96–101.

Smith,A. and Gerstein,M. (2006) Data mining on the web. *Science*, **314**, 1682; author reply 1682.

Smith,A.K. *et al*. (2007) LinkHub: a Semantic Web system that facilitates cross-database queries and information retrieval in proteomics. *BMC Bioinformatics*, **8** (Suppl. 3), S5.

Suomela,B.P. and Andrade,M.A. (2005) Ranking the whole MEDLINE database according to a large training set using text indexing. *BMC Bioinformatics*, **6**, 75.

Tsang,I. *et al*. (2005) Core vector machines: fast svm training on very large data sets. *J. Mach. Learn. Res*., **6**, 363–392.

Williams,K. and Calvo,R. (2002). A framework for text categorization. In *7th Australasian Document Computing Symposium*. Sydney, Australia.

Witten,I. *et al*. (1999) Managing gigabytes: compressing and indexing documents and images. *The Morgan Kaufmann Series In Multimedia Information And Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 519.

Witten,I.H. and Frank,E. (2005) *Data mining: practical machine learning tools and techniques*. Morgan Kaufman, Amsterdam; Boston, MA.