Silverfish: A Contextual Knowledge Extraction and Aggregation System for Academics*

Srinath Srinivasa Aditya Ramana Rachakonda

International Institute of Information Technology, Bangalore 560100, India
{sri, aditya.ramana}@iiitb.ac.in

Abstract

Repositories like arXiv[1] and knowledge bases like CiteSeer[2] are increasingly becoming central to academicians and researchers. However, current systems provide too little semantic support like determining the topic of a paper, modelling the interests of a user, etc. Silverfish is a contextual knowledge extraction and aggregation system that adds a significant amount of semantics to academic repositories. Silverfish automatically extracts pertinent key phrases and models semantic content in the form of a back end cooccurrence graph. This is used for "intelligent" operations like recommendations, finding related content and semantics based routing of announcements.

1 Introduction

Silverfish is a system for academics that caters to the knowledge requirements of various users, such as students, researchers, academic committees, and institutions.

It can be contrasted from related efforts like CiteSeerX[3] or Libra[4] by the added emphasis on knowledge extraction and aggregation. Silverfish does not automatically crawl the web in search of generic academic material. Instead, content on Silverfish is based on material that is actively referred to or uploaded by users. This distinction is analogous to the distinction between enabling search of Web pages by indexing all pages (like Google[5]) and enabling searching of tagged and shared bookmarks from all over the world (like Delicious[6]). So during a crawl, Silverfish only picks up content which would assist the user to further his knowledge.

Users may refer Silverfish to a scientific publication or a course page. Silverfish employs information extraction techniques to identify commonly occurring entities like paper title, author names, institution names, course instructor names and pertinent key phrases. These entities are then maintained in a back end cooccurrence graph using which several semantic associations – both explicit and latent – are extracted. Latent semantics extraction includes finding entities related to a given entity even when they don't cooccur.

Information extraction in Silverfish is somewhat similar in nature to DBLife[7] but the difference lies in the way semantics are modelled in the system, based on the cooccurrence graph. This enables Silverfish to find associations that are not explicitly made. For example, Silverfish may determine that a user is interested in *XML databases* based on the papers bookmarked by the user. Using this, it can inform the user of other related papers on XML databases even if the user has not bookmarked them. Conferences can post their *call for papers* (cfp) which will be routed intelligently to only those users who would be interested in the cfp contents. Such a system can potentially obsolete mailing lists, where users routinely receive several announcements that they are not interested in. Table 1 gives a broad comparison of Silverfish with other related systems.

2 System Architecture

Silverfish collects data through two different channels; user uploads and web crawls. It extracts named entities and key phrases from every document that a user uploads and displays that information back to the user for corrections. This information is presented to the Knowledge Manager which stores that data in the form of a cooccurrence graph for quick and efficient access.

* The authors would like to acknowledge the efforts of the Silverfish team comprising of Abhilash L. L., Dakshina Murthy B. M., Jayanth R., Mohan N., Ritesh M. Nayak, Sivaramakrishna, Varun M. R. and others.

International Conference on Management of Data
COMAD 2008, Mumbai, India, December 17–19, 2008

[1] http://arxiv.org/
[2] http://citeseer.ist.psu.edu/
[3] http://citeseerx.ist.psu.edu/
[4] http://libra.msra.cn/
[5] http://google.com/
[6] http://delicious.com/
[7] http://dblife.cs.wisc.edu/

CiteSeer	CiteSeer is a scientific literature digital library and search engine that focuses primarily on the literature in computer and information science.
DBLife	A system that manages information for the database research community. It automatically extracts structured information from the raw web pages it crawls.
DBWorld	It is a mailing list of users in the database community. It is popular place for the announcements of call for papers of conferences.
Libra	A free computer science bibliography search engine, that allows to search for papers, authors, conferences, journals, etc.
Silverfish	Silverfish is a system of knowledge extraction, aggregation and dissemination for academics.

Table 1: Comparison of different systems of academic information management

The Knowledge Explicator is a set of algorithms that run on the cooccurrence graph that mine for latent associations in the information.

Silverfish allows the user to search for academic documents and uses different ranking techniques based on what is being queried for. It also keeps track of the profile of the user when he uploads documents and when he queries for data. The profile information is used by the Query Manager to automatically send recommendations to the user whenever he logs in. Figure 1 gives the overall architecture of the system.

2.1 Information Extraction

Silverfish extracts information from documents by matching it across already known patterns. Different patterns of information structure are stored in Silverfish to enable this matching. Silverfish runs a document through these patterns and finds a pattern of best fit. It supports three different types of documents viz., publications, course home pages and calls for papers. In publications, the title of the paper, the authors and the abstract are extracted and presented to the user. The user looks for errors and if any, corrects them and gives back the clean information. This clean information is stored in the database. Apart from this, Silverfish uses KEA [1] to extract key phrases from the complete document text and uses them to learn more about the document.

2.2 Knowledge Store

The information is stored as two separate databases using Lucene[8] and MySQL[9]. The Lucene index is a full text index which enables efficient text searching. The information extracted by Silverfish is categorised into several *entity types* and is stored as *entity instances*. A MySQL database is used to store the entity instances which are represented as nodes in a large graph. Edges in the graph primarily represent cooccurrence information. However, several overlay graphs representing other semantics are also used. This is modelled by using different edge types across nodes. There is also the problem of near duplicates among entity instances in the database. As a Lucene index has a good duplicate detection built into it, this is used to detect near duplicates. So, every entity instance in the database is also indexed in a Lucene index for efficient duplicate detection.

2.3 Knowledge Explication Algorithms

Knowledge Explicator is a set of algorithms used to mine latent knowledge in the system. They use the graph of entity instances stored in MySQL. This graph maintains several types of relationships among which the primary relationship is the cooccurrence data. To mine related entity instances Silverfish uses Energy Model [2], which mines synonyms in textual data. Energy Model uses a cooccurrence graph to mine edges between nodes based on the context in which they occur. Another technique by which new edges are mined is by performing a random walk on the cooccurrence graph in the neighbourhood of a certain node and determining a stationary distribution of the visit probabilities. Silverfish uses these techniques to recommend entities of potential interest to users.

2.4 Knowledge-based Routing

Another application of Knowledge Explicator is to intelligently determine which users would be interested in any announcement posted by other users. For example, if a conference organiser posts the *call for papers* of the conference, it should be sent to users who are interested in the conference. Silverfish extracts information from the content posted by users or from the content found on the Web and then this content is stored in the database. Silverfish then finds users who would be most interested in the content and informs them automatically.

2.5 Messaging

Silverfish sends updates, recommendations and notifications to users through messages. So, the messaging component takes care of different formats of messages.

[8] http://lucene.apache.org
[9] http://mysql.com

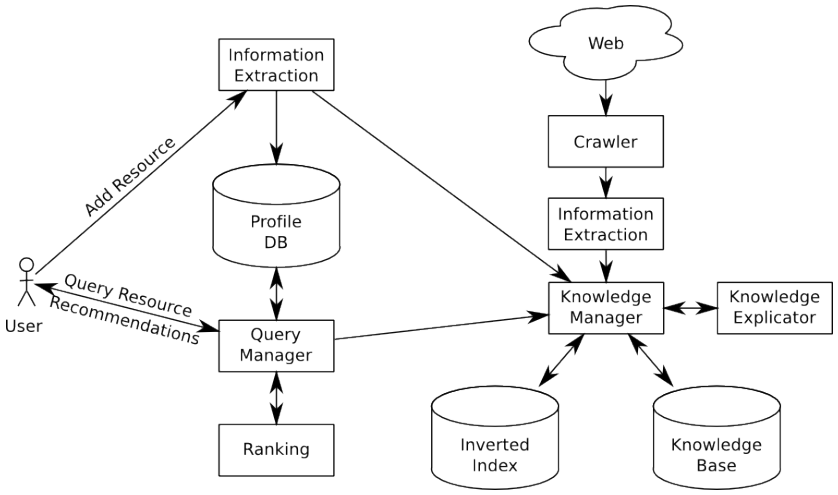

Figure 1: Silverfish system architecture

The messages to a user can be either from another user or can be generated by a recommendation system or can even be system notifications. The notification interface to the user will differ based on the type of the message.

3 Proposed Demonstration

During the demonstration, we would like to introduce the Silverfish architecture, its use cases and its differentiating features. In this process, we would like to demonstrate the utility of such a system to a student, a professor, an institution as well as an organisation. Some of the use cases for demonstration include the following:

3.1 Creating a document collection

Each user can upload documents and Silverfish automatically extracts relevant information from the documents and suggests that to the user. The user can either accept the suggestions or modify them till he finds it fit and save the document to his collection.

3.2 Automated call for papers routing

An organisation conducting an event can upload the *call for papers* of that event and Silverfish will show the different fields related to the event, the user can modify the fields as he sees fit and upload the call for papers. Silverfish then decides the users who would be interested the most in it and intimates them about the event by forwarding the call for papers.

3.3 System recommendations

As the user interacts with Silverfish, the system will build a profile of the user. It utilises this profile to find newer documents unknown to the user and recommends them to the user. This is applicable not just to documents but to other entity types as well. So, users whose areas of interest match are notified of the presence of one another.

3.4 Social network

Users can maintain a list of other users as friends in a social network of users which is built into Silverfish to enable collaboration between users. They can communicate with each other through messages and view the parts of the user profile which are public.

3.5 Latent semantic searches

As mentioned earlier, a person looking for *taxonomy* might be interested in *ontology*. Silverfish actively mines such associations and uses them to retrieve results for user queries. So if a user queries for a topic then documents pertaining to all related topics but not having the user's keyword are also added to the results.

3.6 Course material

Users can add course pages to Silverfish and it will index the course page akin to the way a document is indexed.

3.7 Equivalent entities

If a user finds that two entities are semantically the same he can intimate the administrator about them and the administrator can manually create an equivalence relationship between the entities.

References

[1] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., Nevill-Manning, C. G. (1999). KEA: Practical Automatic Keyphrase Extraction. International Conference on Digital Libraries. California, United States.

[2] Rachakonda, A. R., Srinivasa, S. (2006). Incremental Aggregation of Latent Semantics Using a Graph-based Energy Model. String Processing and Information Retrieval, Glasgow.