Information Retrieval in Life Sciences: The LAILAPS Search Engine

Matthias Lange, Jinbo Chen, Uwe Scholz

Leibniz Institute of Plant Genetics and Crop Plant Research
Corrensstrasse 3 D-06466 Seeland OT Gatersleben
lange@ipk-gatersleben.de

Abstract: Retrieval and citation of primary data is the important factor in the approaching e-science age. Solving the challenge of building a flexible but homogeneous bioinformatics information retrieval infrastructure to access and query the world life science databases is a crucial factor for an efficient building bioinformatics infrastructure.

In this contribution, we demonstrate the use of nine features, which are determined per database entry, in combination with a neural networks as relevance approximator, a novel approach to increase the quality of information retrieval in life science. The implementation of this concept is the LAILAPS search portal. It was designed to support scientist to extract relevant records in a set of millions entries come from private or public databases. In order to consider the fact that data relevance is highly subjective, we support use specific training of several relevance predicting neural networks. In order to make the neural networks working, a continuously training of the networks is performed in background. Here, the system use the user feedback, eighter by conclusions from the user interaction with the query result browser or by manual rating the data quality.

Featured by an intuitive web frontend, the user may search over millions of integrated life science data records. The web frontend comprise a browser for relevance ordered query result, a keyword based query system supporting auto completion, spelling suggestions and synonyms. A data browser is provided to inspect and rate matching data records, and finally a recommender system to suggest closely related records. The system is available at `http://lailaps.ipk-gatersleben.de`

1 Information Retrieval in Life Science

"Getting information is not much of a challenge. Just head for Google, PubMed [Lu11] or Entrez [SEOK96] and get the related web page or database entry." This issue one may get frequently from biologist, if the question be raised which preferred methods or systems are used to get relevant data for a particular biological question [DHW08]. However, getting reliable and relevant information, i.e. to the function of a protein or those proteins that are involved in cancer cell cycle, are much more challenging tasks. The user has the choice of about 1,200 life science databases [Gal12] with billions of database records.

Intuitively, the first choice for information acquisition are web search engines. Web site ranking techniques order query hits by its relevance. However, trying to apply ranking

methods that were developed to rank natural language text or WWW-sites to life science content and databases is questionable [RPB06]. For example, the top-ranked Google hit for "arginase" is a Wikipedia page. This is because the page is referenced by a high number of web-pages or Google assigned a manual defined priority rank. The hypothesis is: A high hyperlink in-degree of a page means high popularity and high popularity means high relevance.

In order to find scientific relevant database entries, scientists need strong scientific evidence in relation to the specific research field. A dentist has other relevance criteria than a plant biologist or a patent agent. The intuitive and commonly used way at the scientist's desktop is query refinement. Criteria like who published, in which journal, for which organism, evidence scores, surrounding keywords are of major importance. Even complete search guides are published, e.g. for dentists [Day01].

Other ranking algorithms use Term Frequency - Inverse Document Frequency (TF-IDF) as ranking criteria. Apache-Lucene[1] is a popular implementation of this concept and is frequently used in bioinformatics, like LuceGene from the GMOD project [ODC+08], which is used for the EBI search frontend EBeye. The TF-IDF approach works well, but misses the semantic context between the database entries and the query.

Andrade and Silva consider the similarity between the result entry and the search query itself as a top-ranking criterion [AS06], while Greifeneder [Gre10] proposes several possible relevance criteria, including the absolute or relative frequencies of the keyword(s) of the search query, the scope or the actuality of the webpage constituting the query result. a website or rather query result.

Another approach is probabilistic relevancy ranking [ILDF07], whereby probabilistic values for the relevance of database fields and word combinations have to be predefined. In combination with a user feedback system, the probabilistic approach shows promising ranking performance [ABD06].

Semantic search engines use methods from natural language processing and dictionaries to predict the semantic most similar database entries. Such conceptual search strategies, implemented in GoPubMed [DS05] or ProMiner [HFM+05], are frequently used algorithms in text mining projects.

2 The LAILAPS Search Engine

In this contribution we apply the LAILAPS search engine [LSB+10] as a system that combines a range of well discriminating database relevance features within a probabilistic model under consideration of user specific relevance profiles. The concept of the LAILAPS information retrieval portal is to provide an information retrieval infrastructure that meets the requirements of the e-science age and to offer an information retrieval platform for data research and exploration. To this end, we built a search engine with the aim to find relevant data in non-integrated life science databases.

[1] http://lucene.apache.org

2.1 The LAILAPS Relevance Prediction

The strategy is to keep a much data structure of the imported databases as necessary to support relevance ranking. But we will integrate data bases at model, schema or data level. Instead, the LAILAPS stores the loaded life science databases in an entity-attribute-value (EAV) adapted database schema. This flexible concept enables the import of RFC-compatible CSV-formatted exports from life science databases, whereas each row comprise a database record and its columns the fields. For the database import, a interactive user interface is provided. For the imported databases, an inverse text index is computed using Apache-Lucene. Furthermore the user may provide synonyms and relevance influencing keywords. In the public available installations, we provide more than 1 million synonyms extracted from the NCBI Entrez system.

The system was designed to provide the look and feel of a web search engine. To support a platform independent implementation and scalable service, we decided to use a JAVA 3-tier web application. The frontend is a web application that supports a keyword based search, a browser for relevance ordered query result, and a data browser to inspect and rate matching data records (figure 2.1). The feedback system enables the user to train the relevance prediction system with individual relevance ratings.

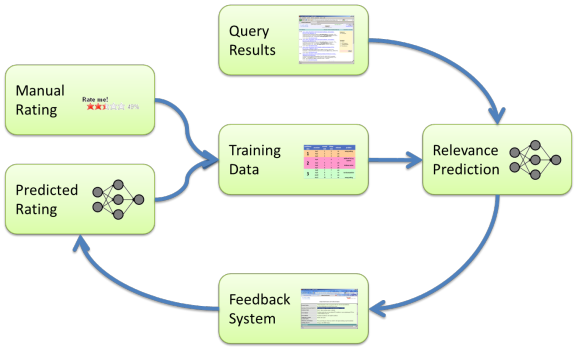

Figure 1: The LAILAPS relevance prediction workflow

The core of LAILAPS is a probabilistic model for relevance prediction on the basis of neural networks. To consider the fact that data relevance is highly subjective to the user of an information retrieval system, we support specific neural networks. Motivated by the observation of user behavior during search engine result inspection, we introduced a set of 9 features. They are well discriminating, and efficiently quantifiable to provide a reasonable fast implementation:

1. attribute in which the query term was found
2. database of the entry
3. frequency of all query terms in the entry and attribute

4. co-occurence, distances and order of the query terms in the entry
5. good or bad keyword near to the query terms
6. the organism to which the entry relates to
7. size of the data section in the entry
8. proportion of the attribute that is matched by the query term
9. whether a synonym expansion was necessary to get the hit

2.2 The Search Engine Software

In order to meet current standards for web information systems and to provide a well scalable implementation to support hundreds parallel user sessions, LAILAPS is implemented as 3-tier system, consist of frontend, business logic and database backend. The frontend is a J2EE web application. The core features are

1. an ad-hoc keyword based query system, supporting auto completion, suggestion and result size estimation;
2. a data browser and feedback system to inspect and rate matching data records.

The business server is implemented as JAVA RMI service and implement the required functions, such as query parsing, synonym expansion, query suggestion, text indexing, feature extraction, relevance prediction, relevance feedback collection. The backend manage the indexed life science databases as well as the text indexes and lookup tables. For this, we use a combination of relational database (H2), key-value database (BerkeleyDB) and inverted index database (Apache LUCENE). This enables LAILAPS to be hosted at single low cost server. Using an 2 core Intel CPU wit 2.4 GHz and a standard SATA HDD, LAILAPS query response time for broad queries with millions of hits (e.g. keyword "gene") in less than 10 seconds. More selective queries take only some milliseconds.

In the matter of fact, user rarely invest time to rate database entries. Rather they inform the search engine indirectly about the relevance of the visited database entry by their behaviour. The obvious reaction to an non-interesting entry is close the page. This and other so called implicite rating are used by the LAILAPS system:

- clicked result entries
- clicked entries above, below
- activity time
- scroll amount
- mouse movement

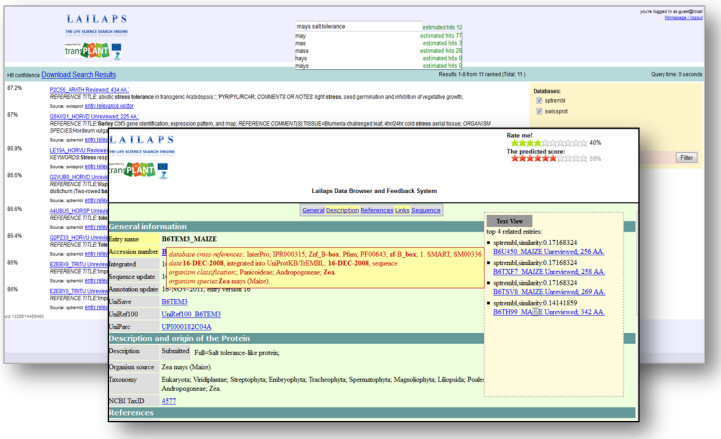

Figure 2: The LAILAPS web frontend – The keyword query submitted as keywords. They are expanded interactively by the query suggestion system. The matching database records are listed according to their relevance. Each record link to its original database and can be inspected embedded in the LAILAPS data browser and feedback system. Here the user may rate the quality and explorer related database entries.

- page lost / got focus
- text selection

Those data is sent to the LAILAPS backend and is used to train the page relevance prediction network. For example, a user rate a page as 80% relevant. While he inspect the data, he like to select, copy past data, and scroll. This is correlated to the given rating. Next time the user show similar browsing behavior, the entry is suggested to 80% relevant. In figure 2.2 the JAVASCRIPT based frontend is illustrated.

Finally, the combination of user relevance rating, explicit and is used to train in intervals the ranking network. Thus, the user has its individual preference for query result ranking.

3 Conclusion

In this paper, we presented the LAILAPS search engine as one promising method for information retrieval in life science databases. Whereas the crucial factor is a relevance order ranking of thousands of database records. The presented concept is to learn ranking pattern from the ranking behavior of scientist, who make use of data retrieval systems. Queries are formulated as simple keyword lists and will be expanded by synonyms. The model is used to extract per database entry a feature vector. Using supervised machine learning approach, we were able to predict a user specific relevance score per feature vector. We the combination of exlicit and implicit user feedback as a promissing approach

for a user relevance feedback.

Supporting a flexible text index and a simple data import format, this concepts are implemented in the LAILAPS search engine. It can easily be used both as search engine for comprehensive integrated life science databases and for small in-house project databases. Using expert knowledge as training data for a predefined neural network or using users own relevance training sets, a reliable relevance ranking of database hits has been implemented. To evaluate the system, a LAILAPS instance with an imported set of UniProt protein data was installed at `http://pgrc.ipk-gatersleben.de/lailaps`

To summarize, by providing portal with the support of individual relevance prediction profiles, the information retrieval has a higher quality and bridge isolated and heterogeneous databases in a non-invasive way. Additionally, the manual information retrieval by query refinement is no longer time-consuming, the scientist can now efficient, with a reproducible quality, pick the data "nuggets" in the data universe. We finally conclude that information retrieval may help to increase the visibility and reusability of scientific data.

Acknowledgment

We thank T. Münch, S. Flemming and H. Miehe as administrator of the project website, Tomcat engine, and Maven developer pipeline. This work was supported by the European Commission within its 7th Framework Program, contract number 283496.

References

[ABD06] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM.

[AS06] L. Andrade and M. J. Sílva. Relevance Ranking for Geographic IR. In *Workshop on Geographic Information Retrieval, Sigir'06*, 2006.

[Day01] J. Day. The Quest for Information: A Guide to Searching the Internet. *Journal of Contemporary Dental Practice*, 2(4):033–043, 2001.

[DHW08] A. Divoli, M. Hearst, and M. A. Wooldridge. Evidence for Showing Gene/Protein Name Suggestions in Bioscience Literature Search Interfaces. In *Pacific Symposium on Biocomputing*, volume 13, pages 568–579, 2008.

[DS05] A. Doms and M. Schroeder. GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Research*, 33(suppl_2):W783–786, 2005.

[Gal12] Galperin, M. Y. and Fernndez-Surez, X. M. The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic Acids Research*, 40(D1):D1–D8, 2012.

[Gre10] Horst Greifeneder. *Erfolgreiches SuchmaschinenMarketing: Wie Sie bei Google, Yahoo, MSN & Co. ganz nach oben kommen*. Gabler Verlag, 2 edition, 2010.

[HFM+05] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck. ProMiner: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.

[ILDF07] Nicholas C. Ide, Russell F. Loane, and Dina Demner-Fushman. Essie: A Concept-based Search Engine for Structured Biomedical Text. *Journal of the American Medical Informatics Association*, 14(3):253–263, 2007.

[LSB+10] Matthias Lange, Karl Spies, Joachim Bargsten, Gregor Haberhauer, Matthias Klapperstück, Michael Leps, Christian Weinel, Röbbe Wünschiers, Mandy Weißbach, Jens Stein, and Uwe Scholz. The LAILAPS Search Engine: Relevance Ranking in Life Science Databases. *Journal of Integrative Bioinformatics*, 7(2):e110, 2010.

[Lu11] Z. Lu. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011, 2011.

[ODC+08] B. O'Connor, A. Day, S. Cain, O. Arnaiz, L. Sperling, and L. Stein. GMODWeb: a web framework for the generic model organism database. *Genome Biology*, 9(6):R102, 2008.

[RPB06] Matthew Richardson, Amit Prakash, and Eric Brill. Beyond PageRank: machine learning for static ranking. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 707–715, New York, NY, USA, 2006. ACM.

[SEOK96] G. D. Schuler, J.A. Epstein, H. Ohkawa, and J.A. Kans. Entrez: Molecular Biology Database and Retrieval System. *Methods in Enzymology*, 266:141–161, 1996.