

# Five Steps to Text Mining in Biomedical Literature

Brigitte Mathiak and Silke Eckstein

Technische Universität Braunschweig,

Institut für Informationssysteme

<http://www.cs.tu-bs.de/idb/>

+49 531 391 3102

{mathiak, eckstein}@idb.cs.tu-bs.de

## ABSTRACT

In this paper we discuss our plans and progress on analysing and integrating various methods of text mining in biomedical literature for a Ph.D project. The framework is a client-based search engine that integrates different machine learning and text mining techniques. For convenience all text mining procedures have been split into five steps, so the different steps can be individually optimised while the rest of the procedures stay untouched. By the use of common interfaces the steps are interchangeable, thus simplifying the objective comparison between them.

## INTRODUCTION

Text mining in biomedical literature has grown in the last few years to be a major tool for bioinformatics. Numerous methods for many applications have been developed. In order to contribute to this still growing field it is important to systemise the methods that are already in use. In our literature research we found that most methods and developments can be divided into five distinct steps. The steps are:

1. text gathering,
2. text preprocessing,
3. data analysis,
4. visualisation,
5. evaluation.

The goal of the thesis is therefore to analyse the different methods applicable to the five steps and to add our own results if possible.

*In Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*, held in Conjunction with ECML/PKDD in Pisa, Italy. 24 September 2004.

The emphasis lies on the computer science focused solutions, as this is the area of our expertise.

As a framework we decided to build a client-based Java search engine which processes query results from various sources including cached previous results into the text mining methods that are currently analysed.

The requirements for the text mining tools to be used in such a search engine are different from normal text mining application as the search engine offers the possibilities of user interactivity. The user gives a first direction in which to find the required documents, thus minimising the number of documents to be checked. Additionally, critical problems can be solved by asking for user feedback. The major restriction, though, on the choice of text mining tools is that they have to compute in near real time or they will lose most of their interactivity advantage. It is therefore necessary to preprocess the information whenever possible and probably use less exact algorithms in order to save computing time.

Furthermore the framework can also be used for more time-consuming problems, bringing together the methods from the different steps, which are therefore easier to evaluate and also easier to integrate. The need for such frameworks has been emphasised throughout the literature [4, 8].

Beside the task of integrating existing methods into the framework and providing interfaces for them, the individual research tasks for each step split up. In text gathering our research focus lies on PDF conversion, as it seems to be a bottleneck in that area and it is also frequently needed for text mining in full text. The text preprocessing research part focuses on how to find the most interesting terms, instead of just excluding the most uninteresting ones. Data analysis research so far emphasises fast document clustering. Visualisation research will most likely concern the visualisation of the document clusters obtained in step three and the transparency of text mining results in general. In evaluation our focus lies on the special evaluation possibilities, like user interactivity.

The paper is structured according to the five steps, each step is discussed in one section. The concluding section summarises our current and future work.

## 1. TEXT GATHERING

The process of text gathering in biomedical literature is largely dominated by PubMed [17], a database containing 12,000,000 references of biomedical publications. The database is searchable by using boolean expressions. It is therefore very easy to obtain large clusters of abstracts, all containing the specified keywords. The most basic step for our search engine is to automatically download the abstracts and to prepare them for further usage. One of our master students implements a program that downloads query results into a specially designed database.

When looking for full papers instead of abstracts, one major source are the 5,800 papers in the PubMed Central Open Access Initiative (PMC OAI) [15]. They have the advantage of being very representative of the overall biomedical literature and have already been transformed to XML, so the structure can be automatically extracted. They proved to be very helpful for comparisons between abstracts and other parts of the text, and also for the evaluation of PDF Conversion tools.

Another application we use them for is to analyse the exact statistical properties of biomedical texts. We are comparing abstracts, full texts and, of course, normal english text statistically in order to find some anomalies that are typical for biological texts, and therefore most likely to be significant to biologists.

Since each area of research develops its own unique vocabulary, the most interesting terms are those that appear in some publications very often and in some just as often as in normal english texts. These terms are likely to be good candidates for topic key words, which are needed in document clustering [10].

The disadvantage of the PMC OAI full texts is that all publications are from a rather small period of time and that they range through many topics, so very few authors appear more than once and there are almost no citation links. For citation analysis a more specialised set of documents is needed [19].

Looking for many full papers with a specialised topic only leaves the option to retrieve the documents directly from the Internet. Google finds 35,000 PDF documents containing the word "arabidopsis" compared to PubMed only finding 4,200 free full texts. Of course, Google search results also contain more useless results, so the actual increase in useful data using Google search remains to be analysed.

When searching for citation links PubMed does not help, as it does not index the full text containing the references section. Using Google or other Internet-wide search engines is required. The strategy of searching the Web with the title of a cited paper or the name or an author has been successfully employed by the citation database Citeseer [7]. The approach showed good results in computer science related publications. Similar success in the biomedical research field seems unlikely, as the ratio of online publications seems to be much less in that area, but nevertheless it could yield interesting results.

Having retrieved the full text from the Internet, it automatically poses the next problem. The full text publications are nowadays usually encoded in PDF [14]. There are conversion tools that convert PDF documents to ASCII or HTML, but these are not as reliable as one would wish for. Especially the graphs and tables are usually not very well converted. To successfully employ text mining on PDF encoded papers, it would be advantageous to start with customising the conversion process in order to be able to

optimise on all levels. Optimising a promising Java open source project so that it performs better in the conversion of scientific publications is part of the work currently done. We are also investigating the possibility to add Optical Character Recognition (OCR) to it, so figures can be adequately mined for useful information.

In the search engine, we strive to integrate all these different means of text gathering, also supporting more unusual approaches like randomised access [22]. The underlying database is supposed to cache the results of all queries made via Internet and also keep track of where individual results originally came from.

## 2. TEXT PREPROCESSING

Text preprocessing classically means tokenisation and then Part-of-Speech Tagging [12] or in a bag-of-words approach word stemming and the application of a stopword list. Tokenisation is the division of text into words or terms. Part-of-Speech (PoS) Tagging tags words according to the grammatical context of the word in the sentence, hence dividing up the words into nouns, verbs, etc. [20] This is important for the exact analysis of relations between words, as it is needed in the extraction of relations between proteins [5].

Another approach is to ignore the order in which the words occur, but instead focussing on the words and their statistical distributions. This is called the bag-of-words approach. In order to use the unordered words it is necessary to index the text into a data vector. The index tends to be very large, so terms that are grammatically close to each other (like "cell" and "cells") are mapped to one term via word stemming and terms that occur very often are removed by compiling stop word lists, so they do not interfere with the data analysis.

So far the Porter's stemming algorithm [16] has been implemented, stop word lists can be dynamically compiled, and basic vector space representations (word frequency and TF IDF from [2]) and similarity measures are implemented.

Yet the goal of text preprocessing is to optimise the performance of the next step: data analysis. The first step of most machine-learning algorithms is to reduce dimensionality by discarding irrelevant data [6]. But it would be even faster if such data had never been gathered in the first place.

As already discussed in the section before, some terms are much more specific for biomedical literature than others, and they are also common enough to be likely mentioned in several different publications. By compiling lists of these specific terms before actually knowing the query, a lot of time can be saved [10].

## 3. DATA ANALYSIS

This is the most diverse and optimised step of the five. Many text mining and data mining techniques are applicable here, as this is where the actual information extraction happens. The data analysis is very dependant on the preprocessing and the data representation model that was chosen in preprocessing. If a vector space representation has been chosen, the data can be analysed using classic data mining techniques, such as support vector machine [11], hidden markov models [18] or artificial neural networks [23], just to name a few. These techniques can be used by the integration of the Weka [24] software package.

In normal search engines, query results tend to be either too large

to process as no person wants to read through 1000 or even thoroughly read 20 publications concerning one topic or the results are to specific to offer any deeper insight into the problem. The application of text mining tools to the results of queries can be a way out of the deadlock of too much and too little information. It may search a given set of papers for protein interactions [3, 9], gene names, etc, and it may also help to decide whether the search results are interesting at all as well as to specify the search. The text mining application for a search engine we plan to conduct some research on is the fast clustering of the results in order to provide assistance to the user by showing him the concepts in the search results already found.

For the task of clustering documents the usual methods to use are unsupervised machine learning, clustering via k-means, SOM or hierarchical clustering. There also exist some variations of these methods that use dimension reduction or Monte-Carlo simplifications in order to save computing time. It remains to be seen how such variations can be applied without trading too much quality for time [1].

#### 4. VISUALISATION

Extracting information that no one sees is useless. So a lot of possibilities have been invented of how to visualise the results obtained. The simplest is just to make a table for the user to look up the information he needs. On the other end of the complexity scala are three-dimensional worlds that the user may navigate in. For the visualisation of query results hypertext is the classic option. The complexity can be hidden, but if the user is interested in the details, he may just click on the link.

The other issue of visualisation is, how much data to show to the user. Usually the user is confronted with pure results without the meta-information on how and why the results were retrieved. This becomes especially important if the results contain some kind of valuation and the user wants to know what it was exactly that made one result superior to another.

The solution to these problems is transparency [21]. Transparency does not mean that the algorithm that made the decision is explained, but the reason for the decision. An example is the Google highlighting of the search keywords in the results. It says: This result was chosen because it has the keyword in it. This is of course a simplification, because the highlighting is just symbolic for the chain of real events that happened, which is in fact quite complicated.

Not all data mining algorithms support this approach, hiding their reasons in neural networks or complex weighting schemes. Further reasearch may improve the situation.

#### 5. EVALUATION

The classic methods of evaluation are the diverse forms of cross-validation and test sets. Supervised machine learning relies on them in order to optimise their parameters. For unsupervised machine learning automatic evaluation is more unusual, since there are normally no evaluation standards. For fast document clustering the results of a slower, more classical taxonomy are suitable and of course the taxonomy of MeSH [13]. It is also possible to do the evaluation on mixed queries, as in the approach from [10]. A mixed query is composed of two keywords only vaguely related to each other, like "drosophila OR human", and

the resultant clusters are then compared to the queries with just one of the keywords.

The search engine framework also brings a rather unusual evaluation criteria: user feedback. By learning which clusters the user chooses, the algorithm may improve itself over time.

#### 6. CONCLUSIONS

In this paper we discussed our plans and progress on analysing and integrating various methods of text mining in biomedical literature for a Ph.D project. We first sketched ideas for a client-based search engine, that will integrate different machine learning and text mining techniques. Our overall goal is to analyse the different methods applicable and to add our own results if possible.

The framework of a client-based search engine provides many possibilities in order to integrate foreign solutions and thus builds the ideal backbone for research in all fields of text mining, as new ideas can easily be tested and their results used to improve further works. It also presents some new tasks on their own, which we have started to explore.

The project is at the very beginning and so far we have concentrated on the first two steps in order to lay out a sound foundation.

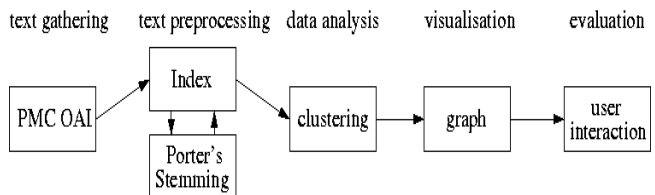


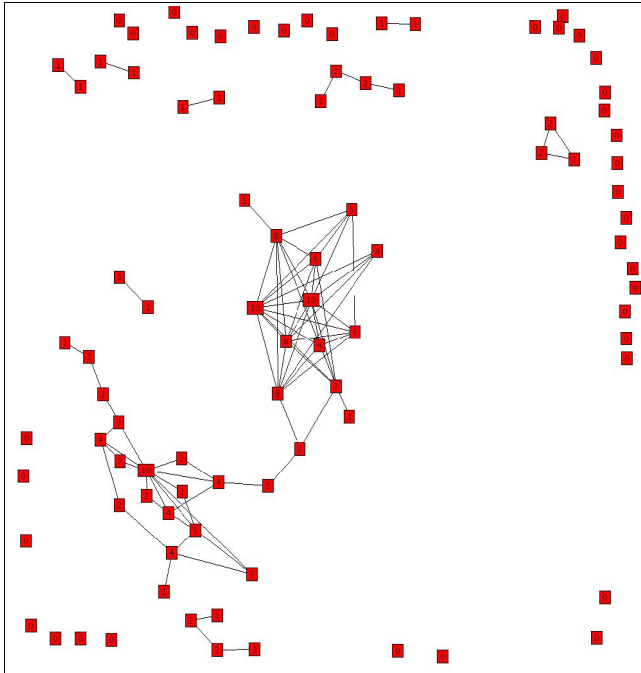
Figure 1 Implemented example of searching topic cluster in PMC OAI full texts

The workflow for one already implemented example for the search engine is shown in figure 1. A subset of the full texts from PMC OAI is converted into an index using a bag-of-words approach. While preprocessing, the words in the index are stemmed. For clustering, a similarity matrix out of the indices is computed. The graph is generated by applying a threshold to the similarity, connecting all documents that are similar enough. The graph is then visualised as depicted in figure 2.

Next, we plan to concretise our plans on fast document clustering. The work we have conducted so far is very promising and we hope to continue with equal success.

#### 7. REFERENCES

- [1] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, J. S. Park, "Fast Algorithms for Projected Clustering", *SIGMOD*, 1999
- [2] R. Baeza-Yates, B. Ribeiro-Neto, "Modern Information Retrieval", Addison-Wesley, 1999
- [3] C. Blaschke, A. Valencia, "Can Bibliographic pointers for known biological data be found automatically? Protein interactions as a case study", *Comparative and Functional Genomics* 2, 196-206, 2001
- [4] S. S. Bhowmick, V. Vedagiri, A. V. Laud, "HyperThesis:



**Figure 2** A similarity graph of 80 documents from PMC OAI

the gRNA spell on the curse of bioinformatics applications integration”, CIKM, 402-409, 2003

- [5] N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, I. Mazo, “Extracting human protein interactions from MEDLINE using a full-sentence parser”, *Bioinformatics*, 20(5), 604-611, 2004
- [6] G. Forman, “An Extensive Empirical Study of Feature Selection Metrics for Text Classification”, *Journal of Machine Learning Research*, 3, 1289-1305, 2003
- [7] C. Giles, K. Bollacker, S. Lawrence, “CiteSeer: An Automatic Citation Indexing System”, *Third ACM Conference on Digital Libraries* 89-98, 1998
- [8] L. Hirschman, C. Friedman, R. McEntire, C. Wu, “Linking Biomedical Language, Information and Knowledge - Session Introduction”, *Pacific Symposium on Biocomputing*, 388-390, 2003
- [9] H. Harkema, R. Gaizauskas, M. Hepple, A. Roberts, I. Roberts, N. Davis, Y. Guo, “A Large Scale Terminology Resource for Biomedical Text Processing”, In *Proceedings of the NAACL/HLT 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for Users*, 2004
- [10] I. Iliopoulos, A. J. Enright, C. A. Ouzounis, “Textquest: Document Clustering of Medline Abstracts for Concept Discovery in Molecular Biology”, In *Proceedings of the Pacific Symposium on Biocomputing*, 384-395, 2001
- [11] A. Kowalczyk, B. Raskutti, H. L. Ferrá, “Exploring Potential of Leave-One-Out Estimator for Calibration of SVM in Text Mining”, *PAKDD*, 361-372, 2004
- [12] C. Manning, H. Schütze, “Foundations of statistical natural language processing”, MIT Press, 1999
- [13] S. Nelson, “Medical Subject Headings – Fact Sheet”, <http://www.nlm.nih.gov/pubs/factsheets/mesh.html>, U. S. National Library of Medicine, 2004
- [14] PDF Specification, [http://partners.adobe.com/asn/acrobat/sdk/public/docs/PDFReference15\\_v6.pdf](http://partners.adobe.com/asn/acrobat/sdk/public/docs/PDFReference15_v6.pdf), Adobe, 2004
- [15] PubMed Central Open Access Initiative, <http://www.pubmedcentral.nih.gov/about/openftplist.html>
- [16] M. F. Porter, “An algorithm for suffix stripping (reprint)”, in *Readings in Information Retrieval*, Morgan Kaufman, <http://www.tartarus.org/~martin/PorterStemmer/>
- [17] PubMed, <http://www.ncbi.nlm.nih.gov/PubMed/>, 2004
- [18] L. Rabiner, “A Tutorial on Hidden Markov Models and selected Applications in Speech Recognition”, *Proceedings of the IEEE* 77 (2) 257-286, 1989
- [19] M. Schroeder, C. Eyre, “Visualisation and Analysis of Bibliographic Networks in the Biomedical Literature: A Case Study”, *Proceedings of the European Workshop on Data Mining and Text Mining for Bioinformatics* 42-50, 2003
- [20] H. Shatkay, R. Feldman, “Mining the Biomedical Literature in the Genomic Era: An Overview”, *Journal of Computational Biology*, 10(6), 821-855, 2003
- [21] D. Szafron, R. Greisner, P. Lu, D. Wishart, C. MacDonell, J. Anvik, et al., “Explaining Naive Bayes Classifications”, TR03-09, Department of Computing Science, University of Alberta, 2003
- [22] S. Schmeier, J. Hakenberg, A. Kowald, E. Klipp, U. Leser, “Text Mining for Systems Biology Using Statistical Learning Methods”, 3. Workshop des Arbeitskreises Knowledge Discovery, Karlsruhe, Germany, 2003
- [23] A. Venkataraman, “Artificial Neural Nets”, <http://www.speech.sri.com/people/anand/771/html/node30.html>, Speech Technology and Research Laboratory, 1999
- [24] I. H. Witten, E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations”, Morgan Kaufmann, 1999