Concept Extraction and Association from Cancer Literature

Yueyu Fu
School of Library and Information
Science, Indiana University,
Bloomington, IN 47405-3907
(812)856-4182, 01
E-mail: yufu@indiana.edu

Travis Bauer
Computer Science Department,
Indiana University, Bloomington,
IN 47405-7104
(812)855-8702, 01
E-mail: trbauer@indiana.edu

Javed Mostafa
Informatics and Information
Science, Indiana University,
Bloomington, IN 47405-3907
(812)856-4182, 01
E-mail: jm@indiana.edu

Mathew Palakal
Computer & Information Science, Indiana University-
Purdue University, Indianapolis, IN 46202-5132
(317)274-9735, 01
E-mail: mpalakal@cs.iupui.edu

Snehasis Mukhopadhyay
Computer & Information Science, Indiana University-
Purdue University, Indianapolis, IN 46202-5132
(317)274-9732, 01
E-mail: smukhopa@cs.iupui.edu

ABSTRACT

There is a large and growing body of web accessible biomedical literature. As this body of electronic literature grows, so does the possibility that document analysis techniques can be used to automatically extract useful biomedical information from them, particularly in the discovery of key concepts dealing with genes, proteins, drugs, and diseases and associations among these concepts. VCGS (Vocabulary Cluster Generating System) was designed to automatically extract and determine associations among tokens from a subset of biomedical literature namely cancer. Such information has notable potential to automate database construction in biomedicine, instead of relying on experts' analysis. This paper reports on the mechanisms for automatically generating clusters of tokens. A formal evaluation of the system, based on a subset of 5338 Pubmed titles and abstracts, has been conducted against the Swiss-Prot database in which the associations among concepts are entered by experts by hand.

Categories and Subject Descriptors

D.2.8 [Database Management]: Database Applications – data mining.

General Terms

Algorithms, Experimentation,

Keywords

Web Data Mining, Web Information Extraction

1. INTRODUCTION

There is now a huge corpus of biomedical literature available electronically, e.g. the abstracts in the Pubmed Database. The complex medical concept relations in this literature are highly valuable. Unfortunately, there are few comprehensive sources of information on biomedicine that explicitly capture and record such associations.

Finding the association among various biomedical concepts is difficult. Most current resources are constructed by hand. Researchers have spent much effort developing systems to automatically mine biomedical literature. Some researchers have focused on detecting gene and protein names [11, 7, 8]. Others put more emphasis on extracting complex interactions among those biomedical concepts [15, 4, 12, 13]. Natural language processing (NLP) techniques are used in these studies. However, the efforts so far have relied on relatively expensive NLP approaches or knowledge bases containing hand-coded rules.

In this paper, we discuss a project aimed at automated database construction in biomedicine. We developed VCGS (Vocabulary Cluster Generating System), a system that automatically extracts and determines associations among tokens from biomedical literature. We used various strategies in the information extraction and retrieval process, both linguistic and statistical.

Our aim in this study is to examine how the key algorithms influence the clustering results and how they interact with each other. A formal evaluation of the system, based on a subset of 5338 Pubmed titles and abstracts, has been conducted against the well-known Swiss-Prot database, in which the associations among concepts are entered by experts by hand. In the future, this system may be used to predict the relations among genes, proteins, diseases, and drugs.

2. TOKEN EXTRACTION AND ASSOCIATION ALGORITHMS

Our implementation of extracting potentially relevant tokens permits the researcher to select specific areas as targets for vocabulary discovery. Also, the researcher can adjust the free parameters in the system to control the granularity (specificity) of the tokens and associations, and to explore the effect of the parameters on the quality of the clusters produced.

WIDM'02, November 8, 2002, Mclean, Virginia, USA.
Copyright 2002 ACM 1-58113-593-9/02/0011…$5.00.

2.1 Token Discovery

The first step in the analysis of medical abstracts is to determine the tokens that are used for further analysis. In other words, a dictionary needs to be created. We use the following technique to select the dictionary.

1. Identify all unique tokens[1]. Only individual words were treated as tokens. Identify the document in which each token appears. Remove from the token list commonly appearing tokens by using a stop-word list.

2. Calculate the frequency of each unique token in each document, the number of documents in which each token appears, and the total number of documents in the training set.

3. Convert the frequency of each unique token/document to a weight based on following formula [14]:

$$W_{ik} = t_{ik} \times \log(N / n_k) \qquad (1)$$

where t_{ik} is the number of occurrences of token t_k in document i. Additionally, $\log(N / n_k)$ is the inverse document frequency of token t_k, where, N is the total number of documents in the training set, and n_k is total number of documents that contain the token t_k. This formula is called

$tf \cdot idf$ (token frequency multiplied with inverse document frequency).

4. Establish a rank for each unique token in each document according to its weight calculated above. For example, the token with the highest weight in a document receives a rank of 1.

5. Sort the list of tokens by rank and token. Extract the tokens that are ranked between $1 - R$ in at least D documents based on the rank and distribution proportion selected by the user. A small value of R ensures selection of highly weighted tokens, and a relatively large value of D ensures that the same token is highly weighted in significant proportion of the training documents.

The essence of the technique above is the weighting scheme in step 3 and the selection of appropriate parameters in step 5. Step 3 ensures that tokens are weighted according to their discriminatory power as measured in terms of their distribution in individual documents and the collection as a whole. The step 3 formula guarantees that tokens with high frequency in a moderate number of documents (important tokens representing a theme in a subset of documents) would be weighted high, while tokens with extremely high frequency in many documents (common tokens in a domain) would be weighted down. Step 5 provides the user further control over fixing the granularity of tokens and the total number of tokens desired. If the user selects low R (rank) and low D (total documents in which token has to be ranked between $1 - R$), highly ranked but specific tokens that occur rarely may be found. Whereas by increasing R and decreasing D, more general tokens that appear more frequently can be found. It is left up to the users to select a combination of parameter values that is appropriate based on their individual needs. As will be shown later, this has a large effect on the quality of the clusters produced.

To measure how many of the tokens extracted are gene names or protein fragments, we validated the tokens against three local databases. One of them is a gene name database containing a list of 84800 gene names, constructed from GeneCards database. We search the database to see if the token matches any of database records. A second one is a protein name database containing 377853 protein names, constructed from Swiss-Prot and TrEMBL protein knowledge base. We search the database to see if the token matches any fragment of protein names. The third one is an English dictionary, constructed from the Webster Dictionary containing 232731 common English words. It is used to filter out common English words from the extracted tokens.

2.2 Utilizing Latent Semantic Information

Having selected the dictionary, the next step is to compute a vector representation for every document. To compute document vectors, we treated each of the terms discovered in the previous step as dimensions in a vector space. Each document was placed at a point in that space by assigning it a value along each dimension. For each term, we used the $tf \cdot idf$ values, computed as indicated above.

To improve the performance of the token association discovery (described in the next section), we wanted to enhance the quality of these document vectors. A process known as Latent Semantic Analysis (LSA)[6, 3, 2, 5], to reduce the rank of the matrix, has been shown to enhance document vectors by using latent semantic structure in the vectors to help eliminate noise and deal with co-occurrence of terms. The term-doc matrix produced using $tf \cdot idf$ is rank reduced according to singular value decomposition. The resulting vectors help to make the implicit latent semantic information explicit.

2.3 Token Association Discovery

It is a straight-forward task to convert a set of vectors describing documents into a set of vectors describing terms. By looking down the columns, one sees the vectors as adjusted by LSA. By looking across the rows, one sees a different vector for each term. We clustered these resulting vectors associated with each term. The process was as follows:

1. Based on the tokens and the associated information generated by the token discovery algorithm, create a machine-readable lexicon file in which each line contains two elements: token identifier and a token.

2. Using the lexicon file produced in step (1) convert each document in the training corpus to a vector V, whereby the dimension of V = number of unique tokens in the lexicon. Each element in a vector would be a weight corresponding to the token for that position derived according to the $tf \cdot idf$ formula presented as equation 1 above.

3. The vectors produced in 2 above form a term-document matrix. From the matrix, term vectors are generated by considering all the weights corresponding to a term as represented in all the documents in the set. To start off the procedure, a term-by-term matrix is produced containing the distances among the term vectors. To calculate the distance the equation 2 below is used. Then the first vector in the set is selected as the initial centroid.

[1] Formally establishing tokens is a complex problem. We employed the definition established in [10] to identify tokens.

$$1-\sum_{i=1}^{t} v_i z_i \sqrt{(\sum_{i=1}^{t} v_i^2)(\sum_{i=1}^{t} z_i^2)} \qquad (2)$$

4. The vector that is farthest from the initial centroid is selected as the next centroid.

5. Compute the distance from each remaining vector to the two centroids and for every pair of the output of the computations save the minimum one. Then select the maximum from the saved minimum distances. If the maximum distance is an appreciable fraction of the distance between the two existing centroids (controlled by a user-selected parameter θ , for e.g., a value of 0.7), then select the vector that had the maximum distance as the new centroid; otherwise halt the procedure and skip the next step.

6. Compute the distance from the rest of the vectors to the existing centroids, and from the output produced for each computation save the minimum for each vector. Choose the maximum from the saved minimum distances and compare it with an appreciable fraction of the typical distance among the three centroids (e.g., average is a good measure). If the maximum is higher then the corresponding vector becomes the new centroid, otherwise halt and go to next step.

Repeat step (6) for the current centroids and remaining term vectors until a maximum distance fails to exceed the appreciable fraction of the typical distance among the existing centroids.

7. Find, for each remaining term vector, the "nearest" centroid and group that vector under the centroid as its cluster member.

The general function of the algorithm is two-fold: (1) to identify tokens that are associated with each other in the corpus in terms of how they group or associate with centroids, and (2) to identify specific groups of tokens, collectively represented as centroids, that are different from each other in terms of their separation as individual clusters. It is hoped that this dual outcome would show for individual tokens their most relevant semantic "neighbors" in single centroids or sometimes more interestingly in multiple centroids.

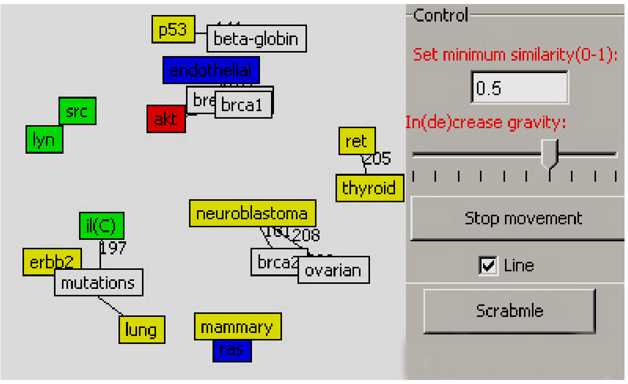
3. AN INTERACTIVE VERSION OF VCGS

The web-based interactive system called LUCAS is a demonstration of the algorithms we described above. It is a servlet running under Apache-Tomcat server in Red Hat Linux 7.2. Users can manipulate various parameters controlling the process explained above for generating clusters: token generation, token validation, and token clustering. The interactive version of this program can be found at: http://lair3.slis.indiana.edu:8080/servlet/sifter.app.cd.vcgs. Screen shot of the interactive system is shown in figure 1.

4. EXPERIMENTS

We conducted experiments to see how well our system could be used to discover relationships among genes and proteins from cancer literature available on the WWW. We considered 5,338 titles and abstracts from the online Pubmed database at www.pubmed.gov. These titles and abstracts were chosen specifically because they contain cancer related gene names. We collected 98 cancer related gene names from http://www.ndsu.nodak.edu/instruct/mcclean/plsc431/cellcycle/cellcycl7.htm which came from the Online Mendelian Inheritance of Man (OMIM). We searched the Pubmed database for those gene names, limiting the searches to the title and abstract fields.

Clusters were created using the methodology outlined above.



Set minimum similarity(0-1): 0.5

☑ Line

Figure 1: Screen shot of interactive visualization of VCGS (LUCAS)

We were interested in measuring how well this system clustered related terms. To analyze the validity of these clusters, we computed overlap. Two terms are said to overlap if they co-occur in an entry in the Swiss-Prot protein database. We computed the average largest overlap in all the clusters returned from the maximin clustering. A large overlap is good because it means that like terms are clustered together. However, since varying parameters in our experiments also varies the number and sizes of clusters, it is not enough to measure only the average size of the overlap. A very large cluster could produce a large overlap, but also contain many terms that are not related. To account for this, the key result we were interested in is the average ratio of the largest overlap to cluster size across all clusters for a given test. A ratio of 1.0 would mean that all of the terms in each cluster are related according to the protein database. A ratio of 0.5 would mean that, on average, one half of the terms in each cluster were related to one another.

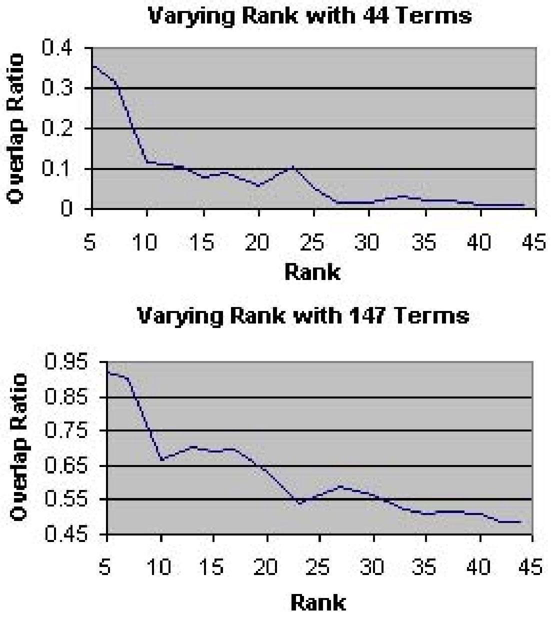


Figure 2: Varying Rank

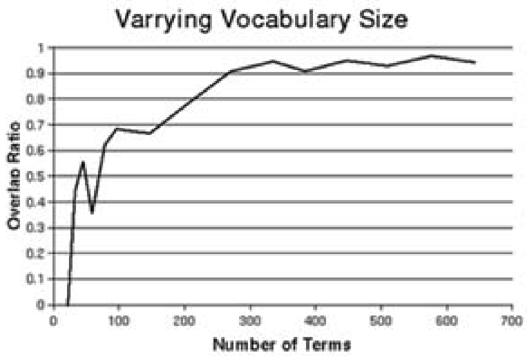


Figure 3: Varying number of terms

We also looked at the effects of varying the number of terms chosen to be in the lexicon. This also had an impact on the overlap ratio. The results are shown in figure 3. What this shows is that a greater overlap ratio is achievable as the number of terms increase. As more terms are included in the term/document matrix, the matrix contains more latent semantic information. Because of this, when the matrix is rank reduced, it is better able to create similar vectors for similar terms.

5. CONCLUSIONS

We have implemented VCGS (Vocabulary Cluster Generating System), a system that automatically extracts and determines associations among tokens from cancer literature collected from Pubmed. Based on VCGS we implemented an interactive web-based association discovery system called LUCAS to aid biomedical researchers to identify useful links among key concepts. We have presented the working prototype of both systems, and investigated the effectiveness of the implemented algorithms in identifying token association. A general finding was that the implemented algorithms are stable, robust, and are capable of providing useful results. A more specific finding was that LSA is an effective technique for extracting relevant associations among terms and that by varying different parameters, one can change the effectiveness of the system.

6. ACKNOWLEDGMENTS

This project was partially supported by an NSF ITR grant #0081944.

7. REFERENCES

[1] Battista, G. D., Eades, P., Tamassia, R., and Tollis, I. G. (1999). Graph drawing algorithms for the visualization of graphs. Prentice Hall, New Jersey.

[2] Berry, M., Dumais, S., and Letsche, T. (1995). Computational methods for intelligent information access. In Proceedings of Supercomputing ’95, San Diego, CA.

[3] Berry, M.W., Dumais, S. T., and O’Brien, G.W. (1994). Using linear algebra for intelligent information retrieval. SIAM Review, 34(7):573–595

[4] Blaschke, C. and Valencia, A. (2002). The frame-based module of the suiseki information extraction systems. IEEE Intelligent Systems, 17(2):14–20.

[5] Caid, W. R., Dumais, S. T., and Gallant, S. I. (1995). Learned vector-space models for document retrieval. Information Processing and Management, 31(3):419–429.

[6] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41:391–407.

[7] Eriksson, G., Franzen, K., Olsson, F., Asker, L., and Liden, P. (2002). Exploiting syntax when detecting protein names in text. In In Proceedings of Workshop on Natural Language Processing in Biomedical Applications.

[8] Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T. (1998). Toward information extraction: identifying protein names from biological papers. In Pac Symp Biocomput, 707–18.

[9] Lewis, D. (1992). Text-based intelligent systems: Current research and practice in information extraction and retrieval. In Jacobs, P. S. (Ed.), Text representation for intelligent text retrieval: A classification-oriented view. Erlbaum, Hillsdale, NJ.

[10] Rindflesch, T., Hunter, L., and Aronson, A. (1999). Mining molecular binding terminology from biomedical text. In Proceedings of the 1999 AMIA Annual Fall Symposium, pages 127–136.

[11] Rindflesch, T., Rajan, J., and Hunter, L. (2000). Extracting molecular binding relationships from biomedical text. In Proceedings of the 6th Applied Natural Language Processing Conference, 188–195.

[12] Salton, G. (1983). Introduction to modern information retrieval. McGraw-Hill, New York.

[13] Thomas, J., Milward, D., Ouzounis, C., Pulman, S., and Carroll, M. (2000). Automatic extraction of protein interactions fromscientific abstracts. In Pac Symp Biocomput, 541–52.