

Biomedical Hypothesis Generation and Testing by Evolutionary Computation

Robert Kozma

Division of Computer Science
University of Memphis, Memphis, TN 38152
rkozma@memphis.edu

Anna L. Buczak

Sarnoff Corporation
201 Washington Road, Princeton, NJ 08543
abuczak@sarnoff.com

Abstract - *Filtering the immense amount of data available electronically over the World Wide Web is an important task of search engines in data mining applications. Users when performing search often formulate hypotheses that they want to find supporting data for. The initial hypothesis reflects their preliminary knowledge of the subject. The final hypotheses at the end of the search reflect what they learned about a given subject and reflect the supporting information they found during search. We propose an evolutionary computation-based method that automatically generates queries and retrieves information to prove or disprove a given hypothesis. In case there is no supporting data, the system evolves another hypothesis for which it can find supporting data. We show preliminary results obtained for hypotheses related to plague where the data used is the Entrez PubMed data set.*

Keywords: Bioinformatics/medicine, Evolutionary Computation, Mining text, Web Mining.

1 Introduction

Efficient data retrieval from large databases and the World Wide Web is an important task that has to be performed routinely in a wide range of applications [1, 2]. In the past decade with the explosive development of electronic data accessible through the World Wide Web, the amount of data available to users became prohibitively large. Often the individual user is not able to process properly the information, creating the potential danger of information pollution.

Information pollution [3] denotes the information overload taken to the extreme. It refers to the fact that the electronic media provides direct access to huge amounts of data at any time, although the given information is often not useful, not relevant, or it could even be harmful to the user by overloading the human sensory and cognitive information processing channels. This can lead to tiredness, fatigue, and degradation of human cognitive performance.

It is highly desirable to develop tools that allow filtering the data by removing the irrelevant pieces of information for a given user and transmitting those which are relevant in a given context of the human decision making. Such tools need to help users finding relevant information quickly in a rapidly increasing volume of available data. It is worth

noting that our biological sensory system has a number of such filtering mechanisms that have developed during our evolutionary process, and presently are embodied in our sensory system. For example our eyes contain optical lenses, which emphasize nearby objects and de-emphasize those far away.

The term *user lens*, directly related to the above filtering needs, has been introduced in the data retrieval literature by Voght [4]. Vogt et al. use that term to emphasize that each user could have their own lens that is employed whenever they utilize the system and is trained with the user's relevance feedback. This lens is a rough model of the cognitive processes of the user when he or she is creating the query or interpreting a document. Our aim is creating such user lenses, to aid human users when interacting with the database in the form of search and data retrieval sequence. Such user lens can be viewed as a supplementary component, add-on, to the biologically evolved sensory system. This filter must include components aimed at both the amount of information and its content. The number of documents retrieved can measure the amount of information. As far as the content is concerned, the design of the filter or lens is more involved. In this work we concentrate on the relevance in the context of a given ontology. Ontology is a conceptualization of a domain into a human understandable and machine-readable format consisting of entities, attributes, relationships and axioms [5]. Ontology uses classes to represent concepts. There are many techniques such as Natural Language Processing (NLP) combined with association rules or statistical models that have been applied to generate ontologies.

Soft computing methods such as genetic algorithms (GAs), evolutionary programming, and fuzzy logic have been successfully used in developing adaptive data categorization and retrieval systems [6, 7, 8, 9]. In the present work we use a GA-based user lens design. The main components of our proposed method can be clearly illustrated by the design of the GA objective function that has two main parts: a quantitative part related to the number of retrieved documents, and a component related to relationships of concepts based on NLP.

The paper starts with a description of our biomedical application, and then continues with the system operation and architecture. Next, we elaborate on the hypothesis representation in a form of a concept map [10];

subsequently the steps of the GA method are described with special emphasis to concept map generation, and evolution. We describe the Chilobot environment [11] that we use to obtain the connection strength between concepts. This is followed by the description of the dataset used. The next section describes details of the implementation in MATLAB software environment, as well as examples of the evolution of queries in the case of actual searches in data obtained from PubMed. Finally, we give conclusions and directions for future research.

2 Biomedical application

A user lens can be useful in various applications dealing with data search and retrieval. The application that we are interested in is biomedical data search, in which the user is looking for information related to a certain disease or pathogen, methods to respond to some biological threat, vaccines and other countermeasures, molecular pathways, drugs leads, etc. In this type of application the main data source of interest is the Entrez database [12]. Entrez provides users with integrated access to sequence, mapping, taxonomy, and structural data. It also provides graphical views of sequences and chromosome maps. A powerful and unique feature of Entrez is the ability to retrieve related sequences, structures, and references. The journal literature is available through PubMed [13], a Web search interface that provides access to over 11 million journal citations in MEDLINE and contains links to full-text articles at participating publishers' Web sites. In this work we use a subset of Entrez PubMed data since we are developing a proof-of-concept approach. Once the system is entirely developed it will be linked to the full PubMed data set.

3 System Operation

The goal of the system is to help users while performing searches of biomedical literature, such as the publications available through PubMed. The full system (Fig.1) comprises a User Modeling Engine, a Feedback-based Hypothesis Adaptation, an Evolutionary Computation Engine, and a data source.

The User Modeling Engine models the user interests based the queries the user submits and the feedback he/she gives to the recommended items. This engine consists of 1) the user model that contains a set of features describing items of interest to the user; 2) a recommender engine that based on user model makes suggestions on new items that are of high interest; 3) machine learning mechanism that adapts the user model and/or the recommender engine to reflect user's current interests and to make more accurate recommendations. Many systems that can perform this task in various domains are described in the literature [14, 15, 16, 17, 2, 18, 19]. As such we will not address this subject any further in this paper.

The Feedback-based Hypothesis Adaptation is a subsystem that works together with Evolutionary Computation and adapts the hypothesis that the user wants to test. The initial

hypothesis can either be inputted as a concept map [10] by the user or can be automatically generated from the user query. A biomedical ontology is used to expand the list of words that can be added to the concept map. The concepts that can be added are synonyms, antonyms, hypernyms, and hyponyms of the words in the initial concept map. The Hypothesis Adaptation algorithm uses the user feedback in order to expand or contract the CM hypothesis.

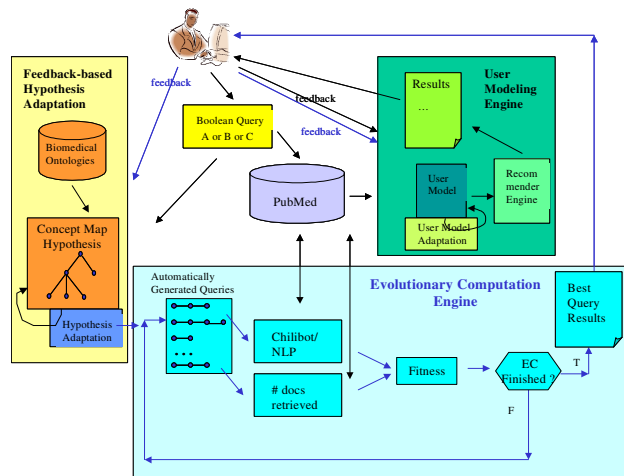


Fig 1. High Level System Architecture.

The Evolutionary Computation Engine (ECE) checks if a given CM hypothesis is true and tries to find supporting data in the PubMed data source. If there is not enough supporting data, the hypothesis is not true with wrt that data source. The most important task of ECE is to generate new hypotheses and judge their correctness. At this point the hypotheses that are evolved and tested are in the form of Boolean queries. The goal of the process is to learn the appropriate concept map that describes proven knowledge about a given topic.

Several biomedical data sources can be connected to the system. At this point the only data source is the Entrez PubMed that allows users to access biological and medical papers.

4 Outline of Hypothesis Generation and Testing Approach

4.1 Hypothesis Representation

A hypothesis in our system is represented as a concept map (CM) [10]. A CM is a diagram consisting of concepts and relations. In Figure 2 an example CM is shown that contains four concepts: *plague*, *medicine*, *virus*, and *bear*. The relations between concepts are presented as lines and usually in CMs they are labeled. So there could be a relation *cause* between concepts *virus* and *plague* meaning to represent the fact (that the user believes in) that a virus causes plague. Similarly there could be a relation *cures* between *medicine* and *plague*. In our proof-of-concept system, the CMs are somewhat simpler: i.e. the relations

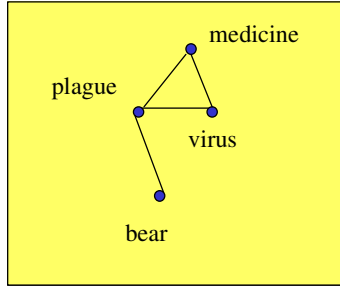


Fig. 2. Example Initial CM.

have no names, it is only important that some relationship exists between two concepts. The CM from Figure 2 represents the hypothesis that medicine, virus and bear are related to plague, and that medicine is related to virus. In the future

work we will be extending the CMs to include relation types (e.g. inhibitory, excitatory).

4.2 Present Simplifications

The aim of the system is to test the initial hypothesis represented as a CM and if not enough supporting data can be found, generate a new hypothesis and find supporting information. The first step to achieve this goal was to develop a simplified system that instead of evolving whole CM graphs deals only with queries. As such our system automatically generates queries and retrieves information to prove or disprove a given query. These queries represent simpler hypotheses to be tested. In the final system there will be an additional module in Hypothesis Adaptation that will link the simpler query-based CMs into a whole CM that fully represents the hypothesis.

4.3 Genetic Algorithm Structure for Hypothesis Generation and Testing

Genetic algorithms (GAs) are stochastic search algorithms based on the mechanism of natural selection and genetic inheritance. They work on a population of individuals that represent potential solutions to the problem. GAs are very effective in finding quasi-optimal solutions to problems with huge search spaces. More information on GAs can be found in [20].

GAs have been used to optimize queries in [6, 21]. We consider queries with a given number of search terms, N . The number of search terms can be an independent parameter in the optimization process. For simplicity here we specify $N=4$, which is a reasonable number in human-directed queries. Obviously, N can be changed and possibly optimized. However, we are interested in other aspects of the optimization process, in particular building relational associations representing hypotheses. Therefore we fix the size of queries for the sake of simplicity. A schematic diagram of the genetic encoding of queries is shown in Figure 3. We use gray-code encoding in the GA and each word (concept) from the dataset that can be used in a query has a corresponding integer number. The operators of crossover and mutation while transforming one integer into another have the effect of transforming one word into another from the dictionary. The GA individual has 5

chromosomes, as depicted on Figure 3. The first four chromosomes encode the query words. The fifth chromosome includes the Boolean operations on the query terms. The encoding allows negation, AND, and OR. When operations of crossover and mutation are performed on the GA individual, each chromosome undergoes them separately, with probability of crossover and mutation being specified per chromosome. We used Stochastic Universal Sampling [22] as the selection mechanism.

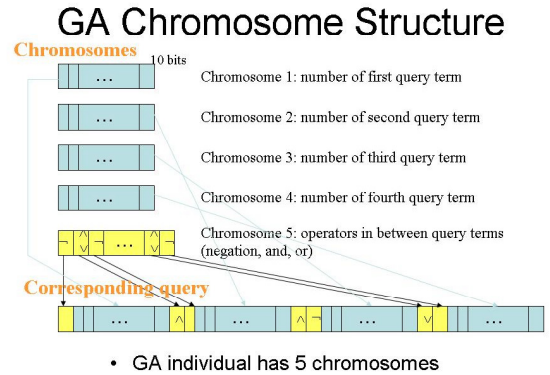


Fig. 3. Schematic of the genetic encoding of a query. Four query terms are used; each term can be negated and logical connectives ‘and’ and ‘or’ are depicted.

4.4 Fitness Function

Optimization is conducted by minimizing the following fitness function:

$$Fitness(Query_k) = w_1 \cdot f_1(NumberOfDocumentsRetrieved) + w_2 \cdot f_2(ConnectionStrength) + w_3 \cdot f_3(GrangerCausality) + w_4 \cdot f_4(DocumentsFromFeedback).$$

The consecutive terms in the fitness function refer to:

1. Quantitative: number of documents retrieved;
2. Connection strength: based on Natural Language Processing for each pair of terms from the query;
3. Causality: Granger causality term for each pair of terms from the query;
4. User feedback: function of the ranking of documents retrieved that user indicated positive feedback for.

In the present studies we elaborate on terms 1 and 2. Term 3 on causality factors represents directedness in the established components in the concept. We suggest using a methodology based on Granger causality [23], and this method will be the objective of our future work. The option to apply user feedback (term 4) has been implemented in our present work. However, we want to demonstrate results obtained by autonomous operation of the system without human interaction and therefore we study the simplified model with $w_3 = w_4 = 0$.

The quantitative component (term 1) represents the optimization factor for queries to produce a desired or ‘reasonable’ number of retrieved documents. In the case of human generated queries, too many retrieved documents are

not desirable, as users can check only the first few pages of the list of documents. Having too few retrieved documents is not good either. Having zero retrieved documents means that the hypothesis has no support in a given data set.

In a simple approach, we aim at retrieving a given expected number of documents (m_0) with a prespecified standard deviation (s_0). The corresponding first term in the fitness function is proportional to a Gaussian distribution as follows:

$$N(m_0, s_0) = e^{-\frac{(m_0 - m)^2}{2s_0^2}}$$

where m is the number of documents retrieved. The values of m_0 and s_0 depend on the given problem and database. Typically the number of retrieved documents m_0 is in the order of a few dozen, and the standard error s_0 is a small fraction of m_0 . Alternative distributions, like a Poisson one, can be also applied. Here we use a Gaussian distribution.

The second term in the fitness function deals with connection strength between each pair of concepts in the query. The relationships between terms are computed using Chilobot [11, 24]. Chilobot is a software robot that uses NLP mechanisms in order to compute relationships between keywords, genes, proteins, etc. It digs into PubMed database and returns the type of relationship between each pair of query terms and its strength. The relationship can be interactive, non-interactive, or abstract co-occurrence only. In case the relationship is interactive its strength is also returned: the strength is between 1 and a maximum value of 30. The higher the number – the stronger the relationship between those query terms. The strength of relationship reflects how many documents were retrieved from PubMed describing this type of relationship between the pair of query terms. Figure 4 shows an example output from Chilobot. When the relationship is interactive, Chilobot further categorizes the relationship as stimulatory, inhibitory, both stimulatory and inhibitory, neither stimulatory nor inhibitory. In the fitness function we use only the strength of the interactive relationship, or 0 if there

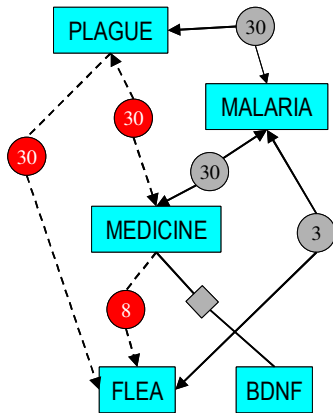


Fig. 4. Example of relationship map obtained by Chilobot. Notation: dashed lines: inhibitory; non-dashed: neither stimulatory nor inhibitory.

is no relationship between terms. In addition to the relationship between the N query terms, we also compute the strength of the relationship between *plague* and each of the terms. We multiply this strength by a factor of α (larger than 1) ensuring that the hypotheses generated by the system are related to *plague* and not any of the other words, e.g. *bear* that might be erroneously in the CM.

The second term in the fitness function is defined as follows:

$$f2 = \frac{100,000}{\alpha \cdot \sum_{i=2}^{N+1} CS(1, i) + \sum_{i=2}^{N+1} \sum_{j=i+1}^{N+1} CS(i, j)}$$

where $CS(i, j)$ is the connection strength (obtained from Chilobot) between i^{th} and j^{th} term in the query, N is the number of terms in the query, $CS(1, i)$ refers to the connection strength between the first term in the dictionary (*plague* in our case) and i^{th} term in the query. α is a constant (we use 4 in our present experiments).

4.5 Document Retrieval

The retrieval of documents corresponding to a given query can use Boolean (crisp) or soft (fuzzy) approaches. Soft queries can produce better results if properly tuned. The main goal of the present study has been to introduce and demonstrate a new methodology of concept learning and adaptation. For simplicity, we use in this work results obtained with crisp queries. Clearly, these results can be further improved, which will be the topic of future studies. We use the standard TF-IDF (Term Frequency Inverse Document Frequency) [25] measure for the terms in the query to obtain the set of retrieved documents.

5 Data Preprocessing

In our project we are interested in developing hypotheses about the *plague*. We obtained plague-related abstracts from Entrez PubMed. The journal literature is available through Entrez PubMed a Web search interface that provides access to over 11 million journal citations in MEDLINE and contains links to full-text articles at participating publishers' Web sites.

We first downloaded all the abstracts from PubMed that were obtained using the query term *plague*. This yielded 2403 documents. The documents ranged from those that described the illness of plague to those that used plague in a much more colloquial fashion.

The next step was to compute for each of the documents the TF-IDF of each word. First we removed the stop words such as *a*, *the*, *is*, etc. Then we used Lucene [26] to compute the frequency of each of the remaining words in each of the documents. Subsequently we computed TF-IDF of each word in each document and that information was stored in a file for consequent use in MATLAB.

We generated a limited dictionary related to the initial concept terms using an existing ontology. We included

synonyms, antonyms, hypernyms, and hyponyms of the initial concept words in that dictionary. The dictionary contained 104 words. Some of those words were plague related (since they were synonyms of *plague*), some were completely unrelated since they were synonyms of *bear* (see Fig. 2). For each pair of words, we used Chilobot to generate the connection strength (or lack of connection information) to be used in the second fitness term.

6 Implementation of Evolutionary Query Optimization

6.1 Computational Environment

We used standard MATLAB environment to implement the design principles described in this paper. The GA implementation is based on the GEATbx Toolbox [27] - in this implementation the lower the fitness function, the better an individual is.

The GA model developed has a large number of parameters. We have conducted extensive studies with these parameters. Here the selected parameters values are summarized. We used a population size of 40 individuals. Optimization at each generation is conducted according to a generation gap of 0.975. We used a crossover rate of 0.7, and the mutation rate of 0.15. We have conducted iterations up to 5,000 generations. Our experience shows that optimum performance has been usually reached much earlier, typically by 2,000 generations, or less. With 104 words in the dictionary for each of the four query terms, the search space for GA is $104^4 = 116,985, 856$. The proof-of-concept system runs in MATLAB and was not designed for speed. It takes about 660 sec to run 2000 generations on a 1.6 GHZ PC.

In the next section we introduce some results of the optimization process, which is interpreted in terms of hypothesis generation and testing concerning the task of *plague* search.

6.2 Evolution of Queries from Initial (Mis-) Concept to Meaningful Final Result

In the experiments performed so far the user wants to test the hypothesis from Figure 2. This hypothesis associates *plague* with *medicine*, *virus*, and *bear*. Obviously the term *bear* is unlikely to be correct even for a non-expert. Also the term *virus* is not correct, since plague is caused by a *bacterium* not a virus. The fact that this initial query represents an incorrect hypothesis allows us to test if the system can rectify the incorrect hypothesis and evolve it into a CM that is meaningful based the knowledge found in the data.

The first two sets of experiments tested GA performance when solely one term was used in the fitness function. Table I shows some of the results obtained when only the first term in the fitness function is used ($w_1 = 1$). All the results are for $m_0=10$ and $s_0=3$ i.e. function has the lowest

value when 10 documents are retrieved, 9 and 11 documents have a higher fitness value, etc. Several optimal results were obtained (rows 4, 6, 9, 10 in Table I) with fitness = 1.0 i.e. they retrieve exactly 10 documents. The results with only f1 used are not sufficient to test the CM hypothesis: there is no way to judge if 6th result “*rodent AND rodent AND antigen AND antigen*” is any more meaningful than the 10th “*plague AND plague AND analysis AND pathogenesis*”. The query corresponding to the initial CM: *plague*, *medicine*, *virus*, and *bear* had a fitness of 205.5 (corresponding to 0 documents retrieved i.e. meaning that the query has no support in the data) much higher than the fitness 1.0 of an optimal query.

Table I. Results for $w_1 = 1$; $w_2 = 0$.

Max gen	Query term 1	Query term 2	Query term 3	Query term 4	fitness	# docs retr
50	plague	infected	Immunization	Immunization	1.65	7
100	serum	capsular	plague	plague	1.06	11
200	black	plague	black	flea	1.06	9
300	treatment	infected	pneumonia	plague	1.0	10
500	plague	pathogen	virus	pathogen	1.25	8
500	rodent	rodent	antigen	antigen	1.0	10
500	plague	antigen	lcrv	vaccine	1.25	8
500	vaccine	lcrv	plague	vaccine	1.06	9
1000	antigen	protein	pneumonia	plague	1.0	10
1000	plague	plague	analysis	Pathogenesis	1.0	10

Table II shows some of the results obtained when only the second term in the fitness function is used ($w_2 = 1$) and the maximum number of generations ranges from 50 to 5000. Each of the results obtained in this case shows terms strongly interlinked among themselves and strongly linked to *plague*.

The results in 9th row are a good example: *plague* is related to *rodent*, to *Yersinia* (*Yersinia pestis* is the agent causing plague), *plague* is also related to *bacteria*, and *death*. So we could say that meaningful CM results can be obtained when using only the second term in the fitness. But there is a problem with that CM: if those words are put in a Boolean query, zero documents are retrieved. Also this method relies too heavily on the results from Chilobot that are sometimes erroneous. Look at 12th query: *bacteria*, *virus*, *rat*, and *Yersinia*. According to Chilobot there is a high interactive relationship between each pair of the words in the query, including *plague*. The relationship between *plague* and *virus* is an interactive relationship with the highest strength 30, that is one of the inaccuracies generated by NLP used in Chilobot. There are many similar inaccuracies, and we want our method to be able to deal with them. This is why we will be using the fitness function that has both terms.

Table II. Results for $w_1 = 0$; $w_2 = 1$.

Max gen	Query term 1	Query term 2	Query term 3	Query term 4	fitness	# docs retr
50	virus	antibiotic	microbe	bacteria	73.21	0
100	Mammalian	antigen	capsular	drug	74.74	0
200	Mammalian	Yersinia	pathogen	virus	73.21	0
300	Yersinia	rat	pneumonic	rodent	73.21	0
500	antigen	rat	bacterium	bacteria	71.74	0
500	rat	rattus	bacteria	treatment	71.74	0
1000	gram-negative	Yersinia	bacteria	rattus	71.84	0
1000	bacterium	infected	rat	bacteria	70.32	0
1000	rodent	Yersinia	bacteria	death	70.32	0
1000	bacterium	Yersinia	bacteria	rodent	70.32	0
1000	Yersinia	bacterium	rat	infected	70.32	0
1000	bacteria	virus	rat	Yersinia	72.89	0
5000	bacteria	bacterium	rat	Yersinia	68.97	0

Table III shows some of the results obtained when using both the quantitative term and the connection strength in the fitness function. In all the results shown $m_0=10$, $s_0=3$, $w_1=1$, and $w_2=1$. The actual set of results was much bigger, including a lot of experimentation with each of those four parameters.

Table III. Results for $w_1 = 1$; $w_2 = 1$.

Max gen	Query term 1	Query term 2	Query term 3	Query term 4	fitness	# docs retr
100	pathogen	infection	human	antigen	85.34	4
200	Streptomycin	pestis	Prophylaxis	treatment	85.34	4
300	human	infection	treatment	pestis	79	10
500	infected	human	bubonic	treatment	78.76	6
1000	human	flea	infection	bubonic	79	10
1000	pestis	antigen	lethal	protein	79	10
2000	antigen	lethal	pestis	protein	79	10
2000	disease	bubonic	treatment	pneumonic	77.39	9
2000	flea	fleas	bacteria	pestis	79	10
5000	treatment	bubonic	human	disease	77.98	7
5000	antigen	vaccine	capsular	pestis	77.58	7

The query based on the initial CM (Fig. 2) resulted in fitness of 334.0457 ($205.5110 + 128.5347$) with 0 documents retrieved. If the system instead of performing the EC, simply performed subqueries of the initial query, the results would be the ones shown in Table IV. Most of those subqueries return no hits or a large number of hits. The few queries that return a reasonable number of results are on various subjects unrelated to the illness of plague

such as avian herpesviruses, bears, history of medicine in China, etc.

Table IV. Subqueries of the query based on initial CM.

Initial Query: *plague AND virus AND medicine AND bear*
0 matching documents

Q: *plague AND virus AND medicine*

3 matching documents (not on the right subject: AIDS, hemorrhagic fevers, avian herpesviruses)

Q: *plague AND virus AND bear*

0 matching documents

Q: *plague AND medicine AND bear*

0 matching documents

Q: *virus AND medicine AND bear*

0 matching documents

Q: *plague AND virus*

396 matching documents

Often articles about influenza

Q: *plague AND medicine*

81 matching documents, e.g.

Unified European higher medical degrees

General internal medicine

Jewish contribution to medicine

History of medicine in China

Q: *plague AND bear*

2 matching documents (not on the right subject)

Q: *virus AND medicine*

3 matching documents (not on the right subject)

Q: *virus AND bear*

0 matching documents

Q: *medicine AND bear*

0 matching documents

The best result obtained so far (8th row of Table III) is “(*plague*) AND *disease* AND *bubonic* AND *treatment* AND *pneumonic*”. The fitness of that query is 77.3919, which is the sum of 1.056 and 76.3359, corresponding to the first and second term in the fitness function, respectively. In this query 9 documents have been retrieved. All the nine results returned provided highly valuable information on plague ranging from the use of plague in bioterrorism, through various types of plague (bubonic, pneumonic, septicemic, asymptomatic, abortive, pharyngeal, and meningial), rapid diagnostic tests for plague, treatment by antibiotics (e.g. streptomycin, cyclones, tetracycline, cholamphenicol, etc.), sub-unit vaccines based on the F1- and V-antigens.

The system was able to automatically generate a meaningful hypothesis related to plague and find very valuable information on plague, starting from a hypothesis that was erroneous. In the future when the system is fully operational, it will work in parallel to the user performing searches on a given biomedical subject, and it will find quickly the best hypothesis that has support in the data set under consideration.

7 Concluding Remarks

In this work we have introduced a system that can help users while they are performing searches of biomedical

literature. The high level architecture of the system was described as well as individual components. Special emphasis was placed on the evolutionary computation method that constitutes the heart of the system and that performs hypothesis generation and testing. This method, starting from the hypothesis provided by the user, evolves and tests new hypotheses, to find the ones that have support in the dataset. An ontology is used to construct a rich vocabulary from the few initial words in the concept map. Each of the words from the vocabulary can be used in the new hypothesis. Each hypothesis generated by the genetic algorithm is tested using a fitness function that takes into account the number of retrieved documents and the connection strength between the terms in the query.

Results were presented that deal with finding hypotheses on the subject of *plague* when using Entrez PubMed data. Our results show that the final hypothesis reflects well the knowledge about the subject of user's interest i.e. *plague*. The supporting data found automatically by the system in Entrez PubMed can be of great interest to the person doing research on that subject. How deep is the information found depends on the data used. At this point, we were using only the abstracts from PubMed - should the full papers be used, the concepts map found would have the potential to contain new information, even for specialists.

Our future work will be directed towards incorporating user feedback whenever it is available. One of the first items that we will address is the development of the full-fledged feedback-based hypothesis adaptation module with automatic retrieval of the concepts from the biomedical ontology. Moreover, we will work on the design of an additional component to the objective function that accounts for apparent causal relationships in the data (Granger causality).

Although the proof-of-concept system described has been shown on a biomedical data set related to plague, the methodology is general and can be applied to any database or data available through the World Wide Web.

8 Acknowledgements

This work is supported by the Defense Threat Reduction Agency and the U.S. Army Medical Research and Materiel Command under Contract No. W81XWH-06-C-0001 awarded by the U.S. Army Medical Research Acquisition Activity. The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of Army position, policy or decision unless so designated by other documentation.

9 References

- [1] Perkowitz, M., and Etzioni, O. (2000) "Towards adaptive Web sites: Conceptual framework and case study," *Artificial Intelligence* 118, 245-275.
- [2] Chen, Z., Meng, X., Zhu, B., and Fowler, R. (2002) "WebSail: From On-line Learning to Web Search", *Knowledge and Information Systems*, 4(2):219-227.
- [3] Nielsen, J. (2000). "Designing Web Usability: the Practice of Simplicity", New Riders Publishing, Indianapolis, IN.
- [4] Vogt, C.C., Cottrell, G.W., Belew, R.K., and Bartell, B.T. (1999) "User Lenses - Achieving 100% Precision on Frequently Asked Questions," *User Modeling UM'99*.
- [5] Quan, T.T., Hui, S.C., and Cao, T.H. (2004) "FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web," *15th European Conf. on Machine Learning/8th European Conf. on Principles & Practice of Knowledge Discovery in Databases, ECML/PKDD, Pisa, Italy*.
- [6] Mitra, S., Pal, S.K., and Mitra, P. (2002). "Data mining in soft computing framework: A survey." *IEEE Trans. on Neural Networks*, 13(1): 3-14.
- [7] Kondadadi, R., and Kozma, R. (2002) "A modified fuzzy art for soft document clustering", *Int. Joint Conf. on Neural Networks, World Congress on Computational Intelligence, WCCI'02, Honolulu, HA, May 2002*, pp. 2545-2549.
- [8] Perugini, S., Concalves, M.A., and Fox, E.A. (2004) "A Connection-Centric Survey Recommender Systems Research," http://arxiv.org/PS_cache/cs/pdf/0205/0205059.pdf
- [9] Kushchu, I. (2005) "Web-Based Evolutionary and Adaptive Information Retrieval," *IEEE Trans. on Evolutionary Computation*, 9(2), pp. 117-125.
- [10] Novak, J. D., and Gowin, D.B. (1984) "*Learning How to Learn.*" New York and Cambridge, UK: Cambridge University Press.
- [11] Chen, H., and Sharp, B.M. (2004) "Content-rich biological network constructed by mining PubMed abstracts", *BMC Bioinformatics*, 5-147.
- [12] Entrez <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>
- [13] PubMed <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>
- [14] Bartell, B.T., Cottrell, G.W., and Belew, R.K. (1995) "Optimizing Parameters in a Ranked Retrieval System Using Multi-Query Relevance Feedback," *J. Am. Soc. Inf. Sci.* 46, 254-271.
- [15] Jansen, B.J. (1997) "Using Simulated Annealing to Prioritize Query Results," *ACM Conf. Computer Science Education*.
- [16] Pazzani, M., and Billsus, D. (1997) "Learning and revising user profiles: The identification of interesting web sites." *Machine Learning*, 27, 313-331.
- [17] Buczak, A.L., Zimmerman, J., and Kurapati, K. (2002) "Personalization: Improving Ease-of-Use, Trust and Accuracy of a TV Show Recommender", *Workshop on Personalization in Future TV, User Modeling Conference, Um'2002, Malaga, Spain, 2002*.
- [18] Alonso, R., and Li, H. (2005) "Model-Guided Information Discovery for Intelligence Analysis", *14th Int. Conf. on Information & Knowledge Management, CIKM'05, Bremen, Germany*.
- [19] Gonzalez, G., Angulo, C., Lopez, B., JL. De la Rosa (2005) "Smart User Models: Modeling Humans in Ambient Recommender Systems," *10th Int. Conf. on User Modeling (UM'05)*. Edinburgh, Scotland, pp.113-122.
- [20] Holland, J.H. (1975) "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor.
- [21] Martín-Bautista, M.J., Amparo-Vila, M., and Sánchez, D. (2002) "Intelligent filtering with genetic algorithm and fuzzy logic," in: *Technologies for Constructing Intelligent Systems*, pp. 351-362, Physica-Verlag, Germany.
- [22] Baker, J. E. (1987) "Reducing Bias and Inefficiency in the Selection Algorithm", *The Second International Conference on Genetic Algorithms and their Application*, (ed.) Grefenstette, J.J. Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates, pp. 14-21.
- [23] Granger, C.W.J. (1969) "Investigating Causal Relations by Econometric Methods & Cross-Spectral Methods," *Econometrica* 37, 424-438.
- [24] Chilobot <http://www.chilobot.net/>
- [25] Salton, G., and McGill, M.J. (1983) "Introduction to Modern Information Retrieval", McGraw-Hill, ISBN 0070544840.
- [26] Lucene <http://lucene.apache.org/java/docs/>
- [27] Pohlheim, H., (1999) "*Evolutionäre Algorithmen - Verfahren, Operatoren, Hinweise aus der Praxis*," with GETAbx toolbox, Springer Verlag, Berlin, New York.