OPTIMIZATION OF KEYWORD GROUPING IN BIOMEDICAL INFORMATION RETRIEVAL USING EVOLUTIONARY ALGORITHMS

Viktoria Dorfer [a], Stephan M. Winkler [b], Thomas Kern [c], Gerald Petz [d], Patrizia Faschang [e]

[a,b,c] Upper Austria University of Applied Sciences
School of Informatics, Communications and Media
Softwarepark 11, 4232 Hagenberg/Mühlkreis, Austria

[d,e] Upper Austria University of Applied Sciences
School of Management
Wehrgrabengasse 1-3, 4400 Steyr, Austria

[a]viktoria.dorfer@fh-hagenberg.at, [b]stephan.winkler@fh-hagenberg.at, [c]thomas.kern@fh-hagenberg.at,
[d]gerald.petz@fh-steyr.at, [e]patrizia.faschang@fh-steyr.at,

ABSTRACT
The amount of data available in the field of life sciences is growing exponentially; therefore, intelligent information search strategies are required to find relevant information as fast and correctly as possible. In this paper we propose a document keyword clustering approach: On the basis of a given set of documents, we identify groups of keywords found in the given documents. Having developed those clusters, the complexity of the data base can be handled much easier: Future user queries can be extended with terms found in the same clusters as those originally defined by the user.

In this paper we present a framework for representing and evaluating keyword clusters on a given data basis as well as a simple evolutionary algorithm (based on an evolution strategy) that shall find optimal keyword clusters. In the empirical section of this paper we document first results obtained using a data set published at the TREC-9 conference.

Keywords: Information Retrieval, Evolutionary Algorithms, Keyword Identification, Document Clustering, Bioinformatics

1. INTRODUCTION

In the year 2009 on average 63.000 new papers per month have been added to PubMed (see www.ncbi.nlm.nih.gov/pubmed), at the moment containing more than 19 million citations. Physicians, biologists, people working in the field of life sciences have to stay up-to-date, regarding for example new therapies, to be able to carry out their work as good as possible. As this huge amount of data cannot be processed manually, intelligent search strategies are necessary to be able to extract as much information as possible for an existing question.

This special field of research is called information retrieval (IR). Many different approaches have already been developed in IR to design intelligent search strategies, including relevance feedback, clustering, parsing and regression analyses (Rocchio 1971; Salton 1988; Schank 1975; Lenat and Guha 1989; Fontaine 1995). The quality of an information retrieval approach can be measured through precision and recall; precision is given by the number of retrieved relevant documents divided by the number of totally retrieved documents, whereas recall is the ratio of the number of relevant retrieved documents and the number of totally relevant documents (see Figure 1). Since it is not trivial to determine the number of relevant documents, certain approximations have to be used.

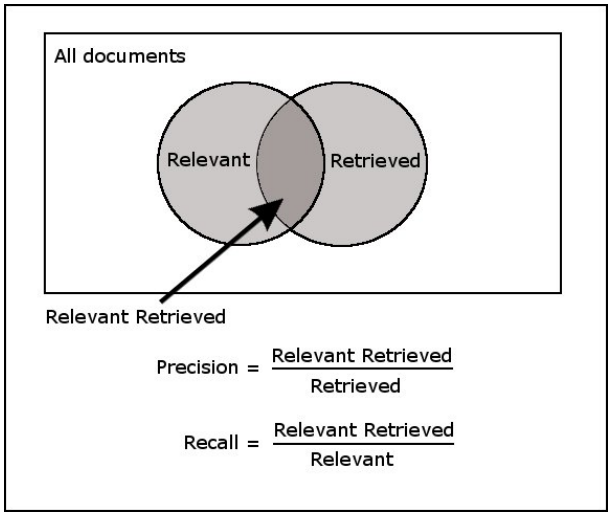

Figure 1: Result Set: Relevant Retrieved, Relevant, and Retrieved (Grossman and Frieder 2004)

However, it is not trivial to find a perfect information retrieval solution (Grossman and Frieder 2004). One reason might be that the approaches were tested with limited data sets, as they were unable to cope with the amount of available data. Still, we cannot make a statement about the performance of a certain approach unless we consider representative data sets (Grossman and Frieder 2004; Hersh 2003).

In this paper we present a method based on evolutionary algorithms that identifies optimal sets of document keywords: The given training sets of documents are analyzed and groups of keywords that are frequently found in combination are generated. Using the produced sets of keywords, future user queries can be updated (extended or transformed) automatically in order to find sets of appropriate documents more efficiently.

2. SCIENTIFIC GOALS

Document clustering is one of the approaches in information retrieval that are used to reduce complexity. Here, documents are divided in groups and the query terms are only matched against the groups assumed to be relevant (Grossman and Frieder, 2004).

Much research has been performed on document clustering, mostly using k-means clustering (Willet 1990; Steinbach, Karypis, and Kumar 2000), but only a few have worked with evolutionary algorithms (Gordon 1991; Robertson and Willet 1994; Jones, Robertson, Santimetvirul, and Willet 1995). Gordon (1991) developed a user based approach where document descriptions are adjusted over time. In contrast, Robertson and Willet (1994) proposed a method to generate groups of words being similarly frequent. Jones, Robertson, Santimetvirul, and Willet (1995) described an approach to search document cluster centers through genetic algorithms.

Apart from the information retrieval scope, work on document clustering based on heuristics has also been carried out, e.g. by Jian-Xian, Huai, Yue-Hong, and Xin-Ning (2009), who worked on a clustering method with known cluster count to optimize classification.

In the context of our research project, the main goal is to identify a document clustering algorithm that is able to cope with a big data base. The idea is to design an algorithm that identifies clusters of documents with respect to sets of keywords that occur in the documents. With these document clusters we now want to adapt queries by adding new relevant terms; these terms are taken from the clusters that seem to be relevant. This approach shall help to improve information retrieval by finding new relevant documents.

In order to reach these goals we have to cope with complex combinatorial problems; we have therefore decided to apply evolutionary algorithms. Details about this approach are given in the following section.

3. APPROACH

3.1. Evolutionary Algorithms

In order to solve complex problems, for which there is no exact and efficient way to find a solution in acceptable time, heuristic methods can be applied. Heuristic methods provide a reasonable tradeoff between achievable solution quality and required computing time, as they employ intelligent rules to scan only a fraction of a highly complex search space. For efficiently scanning complex and exponentially growing search spaces, only heuristic methods can be considered for solving problems in dimensions relevant in real-world applications.

One of the most prominent representatives of heuristics is the class of evolutionary algorithms (EAs): Starting from an (in most cases randomly created) initial population, new solution candidates (individuals) are repeatedly generated by combining attributes of existing solution candidates (which are selected using parent selection operators). These new solutions are optionally modified using a certain mutation routine. The two probably best known types of EAs are the genetic algorithm (GA; Holland 1975) and the evolution strategy (ES; Rechenberg 1973, Schwefel 1994); a detailed overview of evolutionary algorithms, especially genetic algorithms and genetic programming, can be found in Affenzeller, Winkler, Wagner, and Beham (2009).

In the research work described in this paper we have used evolution strategies: Populations consisting of μ individuals are used; in each generation, λ children are generated using random parent selection and mutation. Then the μ best solutions (selected from the children, if the *comma* strategy is chosen, or from parents and children, if the *plus* strategy is chosen) become the next generation's members. We have designed and implemented problem specific mutation operators that are used for modifying solution candidates in order to create new solutions; these operators are described in the next subsection.

3.2. Definition of Solution Candidates

In this work solution candidates represent sets of keywords. These clusters can be initialized either randomly or with keywords based on known statistical information. Mutation can be applied with the implemented problem specific mutation operators: Given a solution candidate c, mutants are created by generating a copy of c and modifying the sets of keywords stored in c. There are several ways how this modification can be done: Randomly chosen keywords might be added to a cluster, keywords of a cluster or even a whole cluster might be removed, a new (randomly generated) cluster could be inserted, or a cluster might be split into two separate clusters.

Neither the total number of clusters nor the number of keywords describing one cluster is fixed. Figure 2 gives a graphical illustration of this idea: Keywords KW_{ij} form cluster C_i; each solution candidate consists of arbitrarily many clusters C_k.

In this context it is very important that the calculated clusters are directly usable: As mentioned previously, the overall goal is to design a framework for biomedical information retrieval which should first cluster document keywords and then use these clusters to extend queries. Provided with the clusters as depicted in Figure 2, it is possible to extend user queries with

terms found in those clusters containing also the original user query terms.

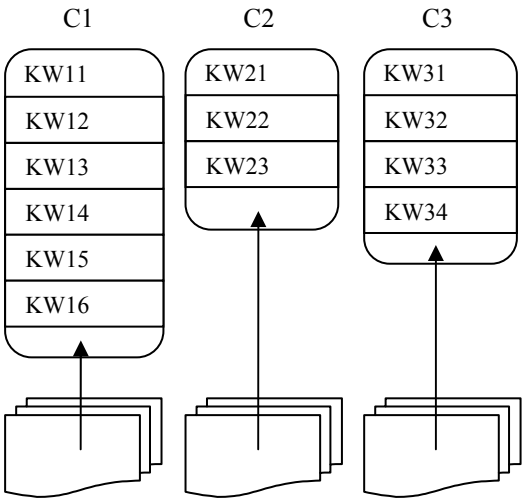

Figure 2: Keyword Clusters Defined by one Solution Candidate

3.3. Fitness of Solution Candidates

The quality of a solution candidate depends on the specificity of the clusters it represents: The better the separation of clusters, the higher the fitness of the candidate. Beside the specificity parts of a cluster we also consider other issues in our fitness function. These aspects cover the number and the distribution of documents assigned to a cluster, or the total number of documents assigned to the clusters. It is also not prohibited that a document belongs to more than one cluster.

Two slightly different fitness functions were tested, each of them considering the following features:

- *A* (range [-∞; 1]): In parameter *A* the ratio of the number of total assigned documents (*At*) to the number of distinct assigned documents (*Ad*) is incorporated; the smaller *A,* the fewer documents are assigned to more than one cluster. In the optimal case the value for *A* would be 1, where every assigned document is assigned to only one cluster.
- *B* (range [0; 1]): Parameter *B* represents the ratio of the number of distinct assigned documents (*Ad*) to the total number of documents (*N*). In the optimal case the value for *B* would be 1, where all documents in the data base are assigned to a cluster.
- *C* (range [0; 1]): Parameter *C* is the mean average cluster confidence, i.e., how good do the documents assigned to a cluster C_i fit to the cluster C_i. The cluster confidence of a cluster C_i ($cConf(C_i)$) is calculated in the following way: For all documents in cluster C_i we calculate the ratio of the number of keywords present in document *d* as well as in cluster C_i, and the number of keywords present in document *d*. These ratios are summed up over all documents in cluster C_i. In the optimal case the value of *C* would be 1, where all documents fit perfectly to their assigned clusters, meaning every document *d* assigned to cluster C_i contains only keywords also present in C_i, whereas C_i can contain more keywords than present in *d*.
- *D* (range [0; 1]): Parameter *D* is the mean average document confidence of each document in each cluster ($dConf(d_j, C_i)$); *D* is a measure how good a cluster fits to a specific document. The document confidence of a document d_j in Cluster C_i is calculated as the ratio of the number of keywords present in d_j as well as in C_i, and the number of keywords of C_i. In the optimal case the value of *D* would be 1, where all clusters fit perfectly to their documents, meaning every cluster C_i contains only keywords which are represented in each document assigned to cluster C_i, whereas document d_j assigned to C_i can contain more keywords than present in C_i.
- *E* (range [-∞; 1]): Parameter *E* considers the standard deviation of the number of documents in the clusters. In the optimal case the value for *E* would be 1, i.e., the same number of documents is assigned to every cluster.
- *G* (range [-∞; 1]): Parameter *G* represents the squared difference of the number of generated clusters to the favored number of clusters (which is defined as a multiple of the logarithm of the number of documents). In the optimal case the value for *G* would be 1, i.e., the number of generated clusters is the same as the number of favored clusters.

$$A = 1 - \left(\frac{At}{Ad} - 1\right) \tag{1}$$

$$B = \left(\frac{Ad}{N}\right) \tag{2}$$

$$cConf(C_i) = \frac{\sum_{d \in d_{C_i}} \frac{|KW:KW \in KW(d)\ \&\ KW \in C_i|}{|KW(d)|}}{\left|d_{C_i}\right|} \tag{3}$$

$$C = \frac{\sum_{i=1}^{CK} cConf(C_i)}{CK} \tag{4}$$

$$dConf(d_j, C_i) = \frac{|KW:KW \in KW_j\ \&\ KW \in C_i|}{|KW(C_i)|} \tag{5}$$

$$D = \frac{\sum_{i=1}^{CK}\left(\frac{\sum_{d_j \in d_{C_i}} dConf(d_j, C_i)}{\left|d_{C_i}\right|}\right)}{CK} \tag{6}$$

$$E = 1 - \sqrt{\frac{1}{CK-1}\sum_{i=1}^{CK}\left(N_{C_i} - \overline{N_C}\right)^2} \tag{7}$$

$$G = 1 - \left(\frac{|CK - \varphi * \log N|}{\varphi * \log N}\right)^2 \tag{8}$$

These parameters yield to the following fitness functions, both to be maximized:

$$F_1 = \alpha \cdot A + \beta \cdot B + \gamma \cdot C + \delta \cdot D + \varepsilon \cdot E + \zeta \cdot G \quad (9)$$

$$F_2 = -(\alpha \cdot \log(z - A) + \beta \cdot \log(z - B) + \gamma \cdot \log(z - C) + \delta \cdot \log(z - D) + \varepsilon \cdot \log(z - E) + \zeta \cdot \log(z - G)) \quad (10)$$

(z has to be greater than 1, thus z:= 1.1 or 2.0, e.g.)

The second fitness function is motivated by the approach of multiplying the logarithm of the punishment factors of each parameter, preventing an unfair high fitness value if only one parameter gets very high (which could be a possible problem when using fitness function F_1).

The optimal fitness value for F_1 is $\alpha+\beta+\gamma+\delta+\varepsilon+\zeta$; the maximum fitness value for F_2 is $-(\alpha \cdot log(x) + \beta \cdot log(x) + \gamma \cdot log(x) + \delta \cdot log(x) + \varepsilon \cdot log(x) + \zeta \cdot log(x))$ where $x=z-1$; in the case of $z=2$ the fitness value of an optimal solution becomes 0 (if F_2 is used).

4. DATA BASES

The data used is part of the ohsumed data file of the TREC-9 conference in the year 2000 and has been used in the filtering track (Vorhees and Harman 2000). To be exact, the file used is "ohsumed89.tar.gz", downloaded from http://trec.nist.gov/data/t9_filtering.html, containing references from MEDLINE (Pubmed 2010) of the year 1989, including title, abstract, MeSH (Medical Subject Headings) terms, authors, source, and publication type. MeSH is the U.S. National Library of Medicine's controlled vocabulary used for indexing articles for MEDLINE/PubMed (NCBI 2010). We extracted 36.890 data sets out of the mentioned data file.

Initially we performed some data preprocessing on the extracted sets for better comparability; we have applied stemming and have removed stop words. Stemming is the reduction of words to their stem. We used a simple stemming algorithm, following the algorithm by Harman (1991), but adapted it slightly: The third rule in the Harman algorithm is to trim words ending with "s", except those ending with "us" or "ss". We added the exception "is" to the rule in order not to trim words like "meiosis" or "synthesis", e.g. The stop word list used was taken from Lewis (2004) to remove common English words such as for example "a", "the", "our", or "usually" in order not to distort the clustering.

5. RESULTS

We tested both fitness functions with various settings, each setting with different numbers of generations (5000, 10000, 20000, 50000, and 70000) and each generation setting was tested 5 times.

Six different parameter settings including different setups for μ and λ have been used in the test series documented here. All in all, the algorithm delivered meaningful results and found useful clusters in different parts of the solution space. Examples for meaningful clusters are the combination of "male", "female", "human", "gov" and "support" or the combination of "spider" and "arachnidism".

Figure 3 gives an overview of the weighting factor settings used in examples 1 to 6, Figures 4 and 5 summarize the results that were retrieved in the respective test series. In test series 1 to 3 fitness function F_1 has been applied, test series 4 to 6 were executed using F_2. μ was set to 20 for examples 2 and 3, and to 1 for all other tests; λ was set to 100 for example 1, to 50 for examples 2 and 3, and to 10 for all other runs.

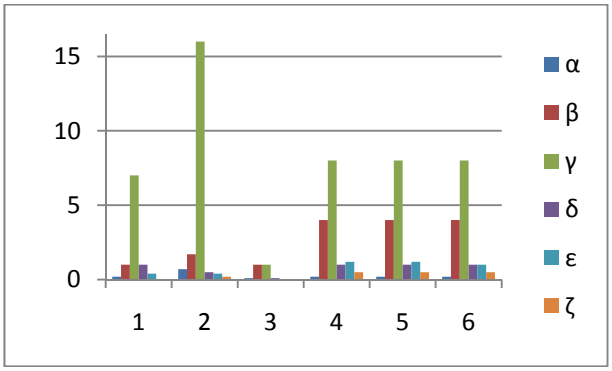

Figure 3: Test Setup (Weighting Factor Settings) of Test Runs 1 – 6

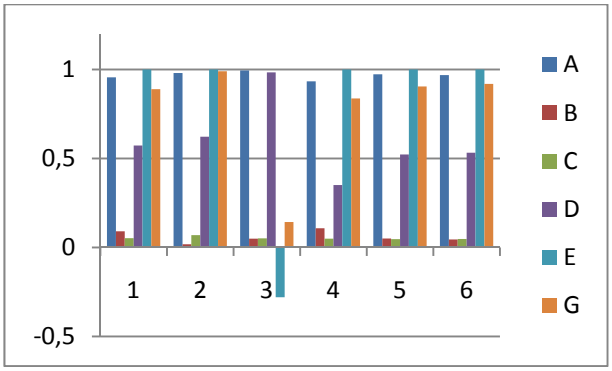

Figure 4: Results of Parameters A-G of Test Series 1 – 6 Retrieved Using Weighting Factors Depicted in Figure 3

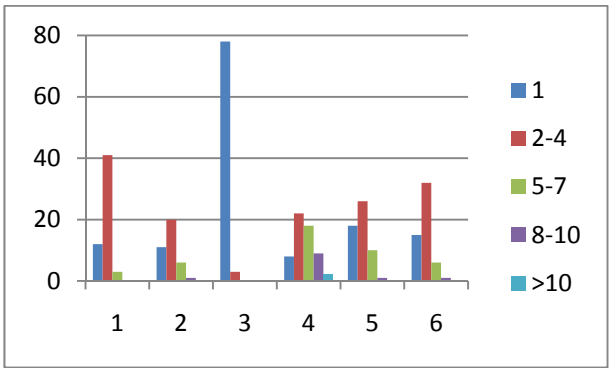

Figure 5: Keyword Histogram for Examples 1 – 6: For Each Example We Give the Number of Clusters Containing 1, 2-4, 5-7, 8-10, or more than 10 Keywords

6. DISCUSSION AND OUTLOOK

6.1. Discussion, Algorithmic Improvements

Even though parts of optimal solutions have been identified, one clearly sees that the presented method is not yet robust if the quality of the given training data set is not ideal (i.e., if the given document keywords are not easily separable into distinct clusters) or the algorithmic parameters are chosen in a suboptimal way. At the moment we suggest combining the results of separate optimization runs and use a combination of their most significant clusters.

As we have not yet been able to find the "optimal" parameter settings, future work will include more parameter tuning and algorithm testing. The authors plan to improve the mutation operators and will also define crossover operators.

6.2. Usage of Text Terms

For future work we also want to use not only the keywords of the documents but all the words of the text for clustering. To filter the significant words we plan to use a commonly used weighting scheme (as described for example in Grossman and Frieder 2004):

$$d_{ij} = tf_{ij} \times idf_j \tag{11}$$

where tf_{ij} is the number of occurrences of term t_j in document D_i – the term frequency – and idf_j is the inverse document frequency, calculated as

$$idf_j = \log\left(\frac{d}{df_j}\right) \tag{12}$$

where df_j is the document frequency, i.e., the number of documents that contain term t_j, and d is the number of documents.

6.3. Implementation

The IR preprocessing approach described in this paper shall be integrated in the HeuristicLab framework (http://www.heuristiclab.com; Wagner et al. 2008; Wagner 2009), a framework for prototyping and analyzing optimization techniques for which generic concepts of evolutionary algorithms as well as many functions to evaluate and analyze them are available. The HeuristicLab framework has been designed and developed by members of the Heuristic and Evolutionary Algorithms Laboratory (HEAL, http://heal.heuristiclab.com/) at the Upper Austria University of Applied Sciences; it is also used as the main framework for the research project *Heureka!* (http://heureka.heuristiclab.com/).

6.4. Biomedical Information Retrieval Framework

As mentioned in Section 2 the authors plan to integrate the clustering algorithm in a biomedical information retrieval framework. This framework shall be used to extend user queries with terms which are members of those clusters also containing the user query terms.

ACKNOWLEDGMENTS

The work performed in this paper was done within *TSCHECHOW*, a research project funded by the basic research funding program of Upper Austria University of Applied Sciences.

REFERENCES

Affenzeller, M., Winkler, S., Wagner, S., A. Beham, 2009. *Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications*. Chapman & Hall/CRC. ISBN 978-1584886297. 2009.

Fontaine, A., 1995. *Sub-element indexing and probabilistic retrieval in the POSTGRES database system.* Master thesis. University of California Berkeley.

Gordon, M., 1991. User-based document clustering by redescribing subject description with a genetic algorithm. *Journal of the American Society for Information Science*, 42 (5), 311-322.

Grossman, D.A., Frieder, O., 2004. *Information Retrieval – Algorithms and Heuristics*. 2nd ed. Dordrecht: Springer

Harman, D., 1991. How effective is suffixing? *Journal of the American Society for Information Science*, 42, 7-15.

Hersh, W.R., 2003. *Information Retrieval: a health and biomedical perspective*. 2nd ed. New York: Springer

Holland, J.H., 1975. *Adaption in Natural and Artifical Systems.* University of Michigan Press

Jian-Xiang, W., Huai, L., Yue-hong, S., Xin-Ning, S., 2009. Application of Genetic Algorithm in Document Clustering. *ITCS '09: Proceedings of the 2009 International Conference on Information Technology and Computer Science*, 145-148. July 25-26, Kiev (Ukraine)

Jones, G., Robertson A.M., Santimetvirul, C., Willet, P., 1995. Non-hierarchic document clustering using a genetic algorithm. *Information Research,* 1(1), Available from: http://InformationR.net/ir/1-1/paper1.html [accessed 1 April 2010]

Lenat, D., Guha, R., 1989. *Building Large Knowledge-Based Systems - Representation and Inference in the Cyc Project.* Boston: Addison-Wesley Longman Publishing Co., Inc.

Lewis, D.D., Yang, Y., Rose, T., Li, F., 2004. A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research.* 5, 361-397. Stop word list available from: http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop [accessed 1 April 2010]

NCBI, 2010. http://www.ncbi.nlm.nih.gov/mesh [accessed 20 July 2010]

PubMed, 2010. http://www.ncbi.nlm.nih.gov/pubmed

Rechenberg, I., 1973. *Evolutionsstrategie.* Friedrich Frommann Verlag

Rocchio, J.J., 1971. Relevance Feedback in Information Retrieval. In: Salton, G., eds. *The SMART retrieval system - experiments in automatic document processing*. Englewood Cliffs: Prentice-Hall, Inc., 313-323.

Robertson, A.M, Willet, P., 1994. Generation of equifrequent groups of words using a genetic algorithm. *Journal of Documentation*, 50 (3), 213-232.

Salton, G., 1988. *Automatic Text Processing*. Boston: Addison-Wesley Longman Publishing Co., Inc.

Schank, R.C., 1975. *Conceptual Information Processing*. New York: Elsevier Science Inc.

Schwefel, H.-P., 1994. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Basel: Birkhäuser Verlag

Steinbach, M., Karypis, G., Kumar, V., 2000. *A Comparison of Document Clustering Techniques*. Technical Report. University of Minnesota, Department of Computer Science and Engineering

Vorhees, E.M., Harman, D.K., 2000. *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC-9)*. Gaithersburg, Maryland: Department of Commerce, National Institute of Standard and Technology.

Wagner, S., Kronberger, G., Beham, A., Winkler, S., Affenzeller, M., 2008. Modeling of heuristic optimization algorithms. *Proceedings of the 20th European Modeling and Simulation Symposium (EMSS2008)*, pp.106-111.

Wagner, S., 2009. *Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment*. PhD Thesis, Institute for Formal Models and Verification, Johannes Kepler University Linz, Austria.

Willet, P., 1990. Document clustering using an inverted file approach. *Journal of Information Science*, 2, 223-231.

AUTHORS BIOGRAPHIES

VIKTORIA DORFER is a senior researcher in the field of Bioinformatics at the Research Center Hagenberg, School of Informatics, Communications and Media. After finishing the diploma degree of bioinformatics in 2007 she was a team member of various projects in the field of informatics and bioinformatics. She is currently working on information retrieval within the TSCHECHOW project.

STEPHAN M. WINKLER received his MSc in computer science in 2004 and his PhD in engineering sciences in 2008, both from Johannes Kepler University (JKU) Linz, Austria. His research interests include genetic programming, nonlinear model identification and machine learning. Since 2009, Dr. Winkler is professor at the Department for Medical and Bioinformatics at the Upper Austria University of Applied Sciences, Campus Hagenberg.

THOMAS KERN is head of the Research Center Hagenberg, School of Informatics, Communications and Media, Upper Austria University of Applied Sciences (UAS). He finished his studies in Software Engineering in 1998. After some work experience in industry he started his academic career at UAS in autumn 2000 as research associate and lecturer for algorithms and data structures, technologies for knowledge based systems, semantic systems and information retrieval. Since 2003 he has conducted several application oriented R&D projects in the fields of bioinformatics and software engineering.

GERALD PETZ is director of the degree course "Marketing and Electronic Business" at the University of Applied Sciences in Upper Austria, School of Management in Steyr. His main research areas are Web 2.0, electronic business and electronic marketing; he has also conducted several R&D projects in these research fields. Before starting his academic career he was project manager and CEO of an internet company.

PATRIZIA FASCHANG received her bachelor degree in electronic business and is a junior researcher at the Research Center Steyr, School of Management, in the area of digital economy. She has worked on several small projects in the field of marketing and electronic business and is currently working on Opinion Mining and Web 2.0 methods within the TSCHECHOW project.