A WEB-BASED USER-PROFILE GENERATOR: FOUNDATION FOR A RECOMMENDER AND EXPERT FINDING SYSTEM

PETR GROLMUS[1]; JIRI HYNEK[2]; KAREL JEZEK[3]

[1]Laboratory for Computer Science, University of West Bohemia, `
Univerzitní 20, 306 14 Pilsen, Czech Republic
indy@civ.zcu.cz
[2]inSITE, s.r.o.
Rubešova 29, 326 00 Pilsen, Czech Republic
jiri.hynek@insite.cz
[3]Departments of Computer Science and Engineering, University of West Bohemia,
Univerzitní 22, 306 14 Pilsen, Czech Republic
jezek_ka@kiv.zcu.cz

The objective of our research is to create a universal tool for recommending non-visited interesting web pages as well as experts working in the same field of specialty. We accentuate practical adaptability of user profiles. User profiles are generated on the basis of Suffix Tree Clustering (STC) algorithm, which is similar to creating an inverted list of phrases occurring in a document collection. We are computing similarity of characteristic phrases identified by STC in order to find *clusters* of phrases. Phrases linked by similarity relationships form a *phrase association graph*. Clusters of phrases generated by our tool define interests of each user. We have tested the system by means of various document collections, such as Reuters Corpus Volume One – RCV1, 20Newsgroups, CTK – Czech Press Agency and Reuters-21578. Experimental results based on our extensive simulations as well as real-life environment are presented in the paper. Precision of our recommender system is 85 to 95 %.

Keywords: text mining; user profile; recommender system; expert search; clustering; suffix tree; phrase search; characteristic phrase; similarity; packet filter.

INTRODUCTION

Information seekers often rely on the web, and web search engines in particular. However, users are often unsatisfied with irrelevant results, and sometimes overwhelmed when hundreds of documents are returned. Mining and searching the web is currently a hot research topic (see, for example, Diligenti et al. [1], Oyama et al. [2], Liu et al. [3], and Yu et al. [4]).

Various research disciplines at universities and other large research institutions attract scientists of different specialty. Because of numerous locations of University buildings combined with overlapping focus of faculties, it is the often case that people of the same interest, working at one institution, do not know each other, which impedes sharing of their experience. It is our objective to match experts based on similarity of their user profiles. Another goal is to recommend new documents that may be of user's interest based on their profile.

To a large extent, our work stems from results presented at Elpub 2002 (Hynek, Jezek: Use of text mining methods in a digital library). Our novel itemsets classifier combined with other classification algorithms is applied to classify artifacts (documents, people) to specific categories (user profiles generated by the system presented in this paper).

It is our objective to utilize work presented at Elpub 2003 (Hynek, Jezek: Practical approach to automatic text summarization), as we can generate summaries of documents being recommended to the user. Machine learning algorithms are applied to select the best sentences (or paragraphs) in

a document, so that user can decide quickly whether a document being recommended to him/her is worth reading.

It is our ultimate goal to have an intelligent system for recommending artifacts of various types, most probably web documents, research papers, persons (people sharing the same professional interests or hobbies), and possibly movies or images.

STATE OF THE ART

The boom of "expert-finding systems", or "recommender systems", started in 1980s and this area is still in focus of many researchers in both commercial and academic world. There are two main reasons to use an expert finding system. First, we may want to find information or an answer to a specific question. In such a case an expert substitutes for a required information source (e.g. a book, a scientific report, etc.). Second, we may want to find a suitable person to solve a given task, e.g. a consultant, an employee, a reviewer, a research worker, etc.

Recommender (expert-finding) systems are often content-based systems, that use machine learning algorithms (classifiers) to recommend items similar to those frequently visited by the user. There are also collaborative-filtering type recommender systems, that identify similarities among profiles of different users registered in the system (profiles are based on attributes specified explicitly by users). Because of some obvious drawbacks, these two categories are often combined into hybrid recommender systems.

Internal systems are used by large companies to recommend experts from their own staff, as opposed to external systems used by smaller companies.

One of the first existing expert-finding systems, called *HelpNet*, was developed by Maron et al. in 1986 [5]. The system responded to user's request by a list of persons, sorted by the likelihood of their ability to answer user's problem. We have recently witnessed several other systems, such as *Expert Finder* (1999) – a hybrid recommender system looking for experts in pre-defined user communities, *WebWatcher* (developed by Lieberman et al.) – sorts and arranges hyperlinks on a web page, *Expertise Recommender*, developed by McDonald and Ackerman at the University of California in Irvine in 2000, RAAP, a hybrid recommender system for document filtering with on-line adaptability of user profiles developed by Delgado in 2000 [6], or *XPERT-FINDER* developed by Sihn and Heeren in 2001 – a system analyzing email communication in order to generate corresponding user profiles.

MRS - Music Recommendation System (Chen [7], 2001) - is designed to provide a personalized service of music recommendation. The music objects of MIDI format are first analyzed. For each polyphonic music object, the representative track is first determined, and then six features are extracted from this track. According to the features, the music objects are properly grouped. For users, the access histories are analyzed to derive user interests. The content-based, collaborative and statistics-based recommendation methods are proposed, which are based on the favorite degrees of the users to the music groups.

DEMOIR (Yimam [8], 2002) - has a modular architecture for expert finding system that is based on centralized expertise models while also incorporating decentralized expertise indicator source gathering, expertise extraction, and distributed clients. It manages to do this by dissociating functions like source gathering, expertise indicator extraction and expertise modeling delegates them to specialized components which can be separately implemented and readily combined to suit an application environment.

Expertise browser (Mockus [9], 2002) - is a tool that uses data from change management systems to locate people with desired expertise. It uses a quantification of experience, and presents evidence to validate this quantification as a measure of expertise. The tool enables developers, for example, easily to distinguish someone who has worked only briefly in a particular area of the code from someone who has more extensive experience, and to locate people with broad expertise throughout large parts of the product, such as module or even subsystems. In addition, it allows a user to discover expertise profiles for individuals or organizations.

OUR APPROACH TO USER PROFILE GENERATION

Information on users' behavior is acquired by means of packet filtering. Packets represent data units used for computer communication on the Internet. Only packets communicating between the client and WWW servers are taken for further processing (are filtered).

The packet filter can be switched off any time, e.g. to prevent undesirable alteration of user's profile by some documents, or to maintain user's privacy. User profiles are generated with the aid of Suffix Tree Clustering (STC) algorithm, which is similar to creating an inverted list of phrases occurring in a document collection. Complexity (both in terms of time and memory requirements) is in the order of $O(n)$, where n is the number of documents in the collection. All documents displayed in user's web browser are passed to STC, upon being processed by our stemming and stop-word filtering engines. Please note that stemming is a process of transforming words into their basic form – stem. Stop-words are such words which can be ignored in keyword-based queries without a significant impact on retrieval accuracy. In the case of document clustering, stems do not play an important role either, and therefore can be deleted from clustered documents.

Before we proceed to text stemming, it is necessary to identify document language, as stemming process is highly language dependent. We cannot assume, however, that the whole collection is written in one language only. An ordinary user may have limited ability to understand several languages. That is why we have developed a technique for document language identification based on occurrence of stop-words. We currently recognize several languages such as Czech, English and German. First of all, we make a summary of all stop-word occurrences and then we figure out the ratio for all languages. If more than 70 % stop-word occurrences fall within a specific language category, we suppose that the document is written in this language. On the other hand, if our limit is not reached for any of the languages, the document is not passed to further processing to minimize errors. Language identification is fundamental for stop-words removal and lemmatization. There are some stop-word homographs in different languages (i.e. Czech conjunction "a" and English indefinite article "a"). It is quite difficult to differentiate among languages with the same roots, such as Slavonic languages of Czech and Slovak, for example.

SUFFIX TREE CLUSTERING

Suffix trees date back to 1970s (Wiener [10]; McCreight [11]). It took many years for suffix trees to gain recognition and wide usage. Unfortunately, suffix trees are relatively greedy in their space requirements. Nowadays, the point of view at suffix tree has changed, because of significant advancements in computer hardware.

Suppose that *lmax* is the maximum phrase length. We can create suffix tree as follows:

1. Tree initialization (tree contains root only);

2. Let's get first *lmax* words *w1…wlmax* from input document (or less, if there are no more words);

3. Each input word wi, where *i=1,…,lmax,* is inserted to STC tree level *i*, so that path from the root of STC tree to the node inserted most lately goes through nodes associated with words *w1…wi-1*. We must save document number for each node (i.e. phrase) to keep track where a specific phrase is used;

4. Remove the first word from the input and if there are other words to process, continue with step 2;

5. Use next input document, continue with step 2.

Now let's create suffix tree for the following three sentences (documents) and find out characteristic phrases (Zamir, 2003 [12]).

Let's have three documents:

1. Cat ate cheese.
2. Mouse ate cheese too.
3. Cat ate mouse too.

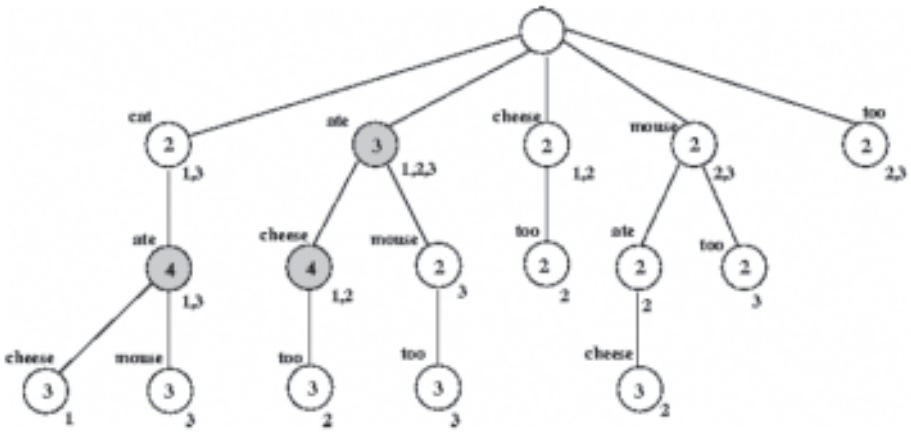

FIGURE 1: EXAMPLE OF AN STC TREE

Weight of each phrase in STC tree (see numbers inside of nodes in Figure 1) is defined as follows:

$$w(f_i) = L(f_i) \times N(f_i)$$

where parameters are defined below. The numbers attached to nodes represent numbers of documents, where the given phrase appears. If we define conditions for characteristic phrase selection as *w(fi) >3 and N(fi)>2*, we would select these characteristic phrases: *cat ate; ate; ate cheese*.

In order to accentuate the importance of long phrases occurring less frequently and improve precision by involving TFxIDF approach, the following formula is used in our system:

$$w(f_i) = L(f_i)^2 \times N(f_i) \times \sum_{j=1}^{m} (tf_{ij} \times \log \frac{m}{df_i})$$

where *L(fi)* represents the length of phrase *fi* (expressed in significant terms), *N(fi)* is the

number of occurrences of such a phrase, *tfij* is the number of occurrences of phrase *fi* in document *dj*, *m* is the total number of documents in collection, and *dfi* is the number of documents containing phrase *fi*.

In order to save computing time when re-creating STC tree, we maintain a list of documents being processed and precise time of their visit. When re-creating the tree, we neglect links to all documents that are already out of "time window". The tree hereby rebuilt includes all documents visited since the last tree processing. Consequently, clusters characterizing new user profile are identified.

Looking at STC tree we have just created, we select *p* characteristic phrases of the highest weight. The resulting number of characteristic phrases is defined as follows:

$$p = r \times \frac{s}{m} + \frac{m}{k},$$

where *m* is the number of documents, *s* is the total number of word positions in all documents, *s/m* thereby being average number of words in a document, *r* is a constant corresponding to proportional representation of the average document length with respect to the total number of words *s*. The purpose of constant *m/k* is to increase the number of characteristic phrases by one for each k documents in the collection.

We are computing similarity of characteristic phrases identified by STC in order to find *clusters of phrases* (using a mechanism often used to analyze contents of shopping baskets in supermarkets, called "frequent itemsets mining"). Similarity between two different phrases is identified on the basis of a set of documents containing both phrases:

$$\frac{|D_m \cap D_n|}{|D_m|} \geq \phi \; AND \; \frac{|D_m \cap D_n|}{|D_n|} \geq \phi \;,$$

where D_m and D_n represent documents containing phrases f_m and f_n, respectively. *f*Ó represents a threshold with a significant impact on cluster generation. The longer phrases are used for user profile identification, the lower threshold ϕ for similarity metrics must be selected (and vice versa).

Phrases linked by similarity relationships form a *phrase association graph*. *Profile clusters* are represented by *maximum subgraphs* of the phrase association graph. The number of these subgraphs represents the number of clusters (i.e. different areas of user's interest). The first practical tests indicate that it is reasonable to believe that one user will not be characterized by more than three or four identifiable areas of interest.

Here is an example of some clusters representing an association graph.

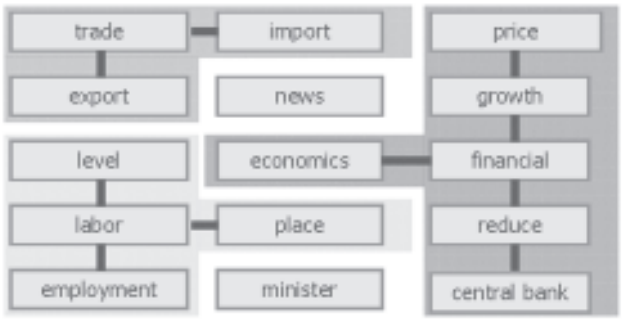

FIGURE 2: AN EXAMPLE OF AN ASSOCIATION GRAPH

It is essential that user profiles are adaptable. User interests vary over time. In order to make a profile "live" with user, the system must accept changes in user's interests. Adaptability of user profiles is ensured by re-generating profiles using documents visited in a specific period only ("time window"). Page view frequency and link visit percentage, among other parameters, are used for this purpose. Adaptability of user profiles could be maintained by means of a persistent data structure, so that each profile corresponds to user's web activity in the past six months (or any other pre-defined period). Potential alterations of web page contents at already-visited web addresses are also taken into account. Profiles are regenerated at regular intervals (incrementally), short enough to meet browsing and expert-matching needs of our users (such as one month).

SYSTEM'S DESIGN

Information on users' behavior is acquired by means of packet filtering. Our packet filter captures all user requests for WWW documents – packets distributed via port 80 (standard HTTP port) and stores them in a database for further processing (step 1 in Figure 3). It is great advantage of the packet filter that it can be switched off any time, e.g. to prevent undesirable alteration of user's profile by some documents, or to maintain user's privacy. Users of our system are identified by randomly generated 32-byte strings.

After collecting certain number of packets in our database (i.e. hundreds or thousands URL requests), the system gathers these documents and generates user's document collection (step 2).

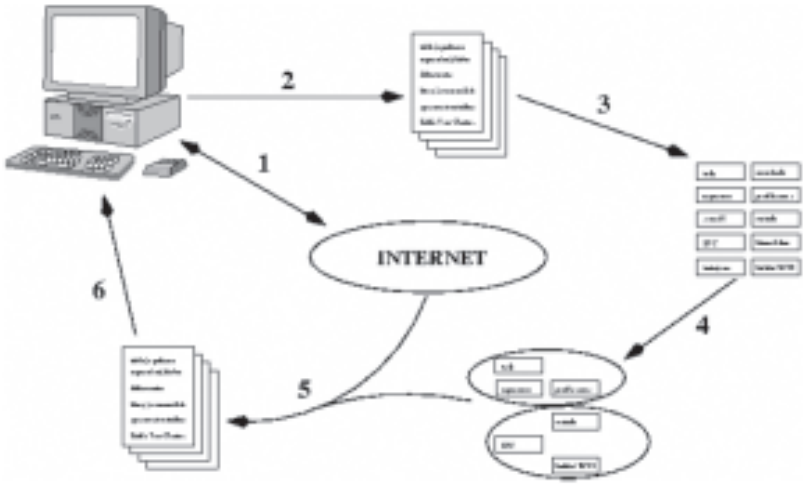

FIGURE 3: GENERAL DESIGN OF PROFILE GENERATOR

User profiles (clusters representing various user's interests) are generated from document collection with the aid of STC algorithm (steps 3 and 4 in Figure 3).

Upon forming clusters as described above, we can use these in variety of applications, e.g. in co-operation with a search engine (step 5) to recommend interesting documents not yet visited by the user (step 6). Other possible applications include finding domain experts, query search expansion, foundation of virtual communities (collaborative filtering), search ranking with respect to user's profile, etc.

The following novel approaches are utilized in our work:

. Document cleaning by means of a sophisticated stemming algorithm

. Computation of phrase weights - modified TF•~IDF

. Idea of phrase centroids to identify user's interests

. Applications – web page recommender, expert finder, query expansion, search ranking, web page pre-fetching

. Implementation in real-world environment (University intranet)

. Web Proxy server is not utilized (independent of IP address; browsing time cannot be measured)

. Relevance feedback is not used (can be counter-productive, as users might feel annoyed by evaluating web forms).

PRACTICAL EXPERIMENTS AND RESULTS

We have recently monitored behavior of selected users (mostly PhD. students) connected to Internet via University network. Sets of characteristic phrases were identified for each user by applying STC. Clusters of phrases generated by our tool define interests of each user. After having defined these clusters, we can start building up a social network spanning across University users to match people of similar interests or expertise. Users are associated with a specific social network based on similarity of phrase clusters (characterizing each user), thereby creating a cluster association graph. Each maximum subgraph of such a cluster association graph represents a specific social network. It is expected that a large number of different social networks will be identified because of natural diversity of the academic environment.

Before collecting sufficient volume of testing data, we have tested the system by means of various document collections, such as Reuters Corpus Volume One – RCV1 (more than 800 thousand documents), 20Newsgroups (20 thousand), and CTK – Czech Press Agency (131 thousand). Main attributes of these collections are depicted in Table 1.

TABLE 1: OVERVIEW OF TESTING COLLECTIONS

	Collections		
Attribute	CTK	RCV1	20Newsgroups
Number of documents	130,955	806,791	19,997
Number of words	29 mio	193 mio	5.23 mio
Average document length (number of distinct significant terms)	159.0	88.4	155.0
Shortest / Longest document (in words)	10 / 5 721	10 / 3 996	1 / 6 695
Avg. number of topics a document is classified to	1.73	3.2	1.0
Number of topics	42	103 (79 leaves)	20

Classification in RCV1 is hierarchical (tree structure), CTK and 20 Newsgroups are non-hierarchical (flat structure).

Topmost level classes in RCV1 collection include *Corporate/Industrial, Economics, Government/Social,* and *Markets*. Main CTK classes include *Politics, Sport, Companies,* and *Police & Law*.

CTK collection, provided by the Czech Press Agency, is the only Czech collection of practical use. It consists of 1999 news archive.

Experiments were performed as follows: Two thirds of documents from selected topics were used to generate clusters (they substitute documents visited by user). Remaining 1/3 was mixed with the same number of non-relevant documents from other topics. This set represents documents returned by the search engine. Similarity of document *D* and cluster *Ci* is defined by means of cosine metrics:

$$Sim(C_i, D) = \frac{\sum_{h=1}^{H} (w_h \times d_h)}{\sqrt{\sum_{h=1}^{H} (w_h)^2 \times \sum_{h=1}^{H} (d_h)^2}},$$

where *H* is the number of characteristic phrases of cluster *Ci, wh* is the weight of *h*-th phrase, and *dh* is the number of occurrences of *h*-th phrase in document D. If *Sim(Ci, D)*.*f*Ñ, where *f*Ñ is a specific threshold, then *D* • ¸*Ci*, which means that document *D* is considered relevant with respect to user interest area *Ci*.

With the a priori knowledge of document categories in tested collections, we can evaluate precision *P* and recall *R* of profiling-process modeling:

$$P = S /(S + S),\ R = S /(S + N),$$

where S_r represents selected relevant, S_n are selected non-relevant, and N_r are non-selected relevant documents.

Tables 2 and 3 show results of profile-modeling experiments. Threshold 1 τ represents document vs. phrase similarity.

TABLE 2: RCV1 RESULTS

Recall							
Threshold τ	0.5	0.6	0.7	0.8	0.85	0.9	0.95
Result	83.54	75.05	55.59	42.75	33.18	27.02	19.25

Precision							
Threshold τ	0.5	0.6	0.7	0.8	0.85	0.9	0.95
Result	57.09	62.61	72.08	83.35	88.41	92.39	93.00

TABLE 3: CTK RESULTS

Recall							
Threshold τ	0.5	0.6	0.7	0.8	0.85	0.9	0.95
Result	97.60	94.60	82.80	63.61	54.01	44.35	32.69

Precision							
Threshold τ	0.5	0.6	0.7	0.8	0.85	0.9	0.95
Result	57.47	61.55	66.47	85.22	85.28	89.99	90.34

Results for CTK collection are depicted in a graph in Figure 4. Please note that precision is much more important than recall in our application, as we prefer recommend a smaller number of truly relevant artifacts (higher precision), rather than higher number of potentially irrelevant ones.

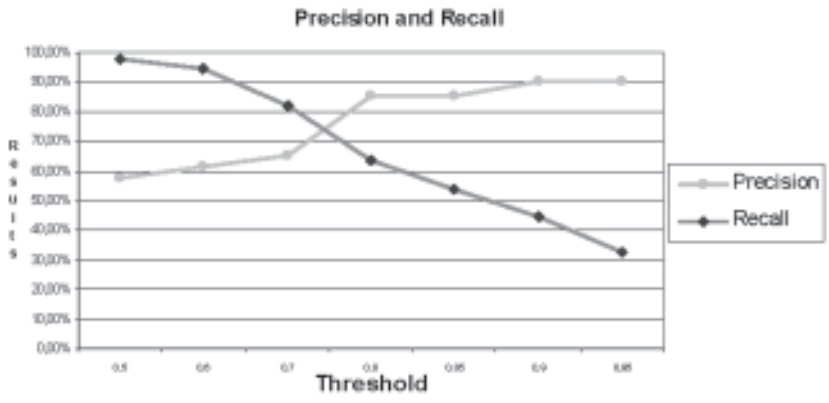

FIGURE 4: PRECISION AND RECALL FOR CTK COLLECTION

Here are clusters C1 to C6 generated on the basis of CTK news (translated from Czech to English for this purpose):

C1: Iraq, Iraqi

C2: Belgrade, Yugoslav, Kosovo, liberation army, Serbian, Albania, Kosovska Albania

C3: Israel, Israeli

C4: press conference, conference, press

C5: Moscow, Russian, Russia

C6: foreign, foreign minister

(Docs: 6 362; words: 1 003 072; STC nodes: 1 397 413)

Figures 5 and 6 below illustrate the impact of phrase length on precision and recall. In the case of Single-term curve, we neglect occurrence of multi-term phrases. If we emphasize the importance of phrase length (Multi-term 2), we achieve higher precision at the cost of lower recall.

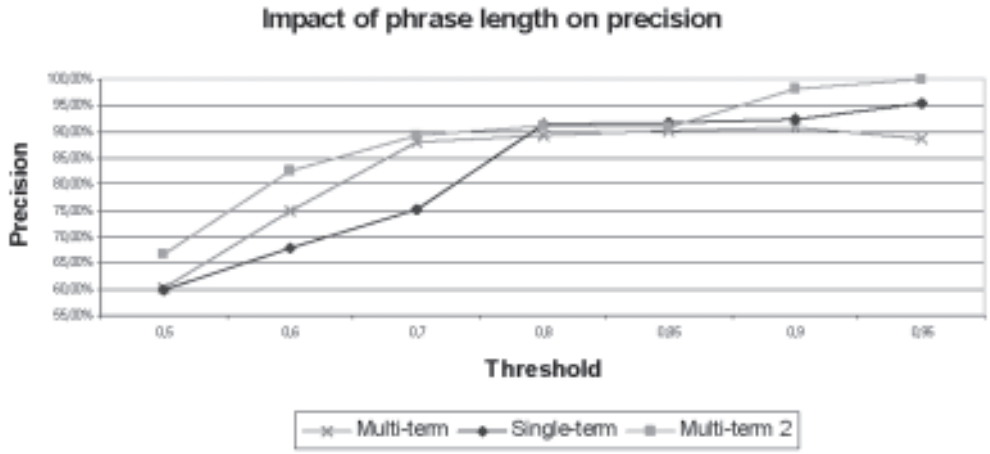

FIGURE 5: IMPACT OF PHRASE LENGTH ON PRECISION

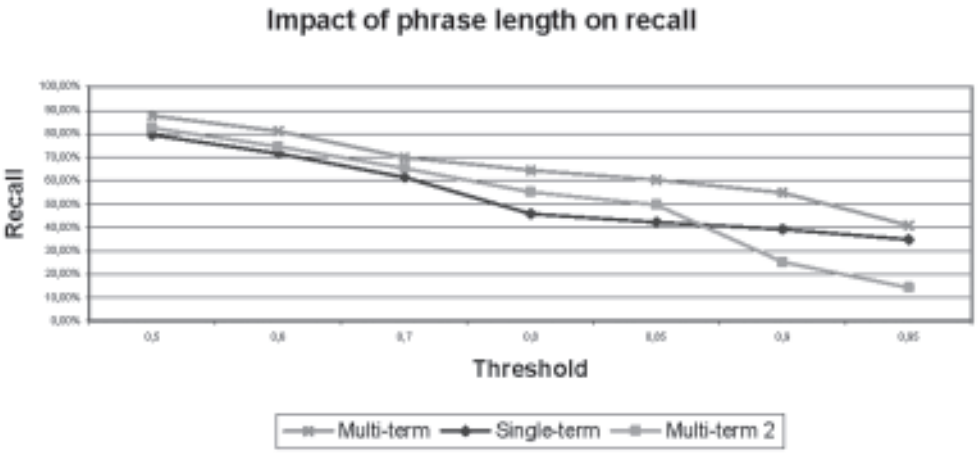

FIGURE 6: IMPACT OF PHRASE LENGTH ON RECALL

CONCLUSIONS AND FURTHER RESEARCH

Presently there are several PhD. students testing the system. Here is an example of a profile generated for one of these students:

C1: image, generate, language generate, research, information, base, language, natural, system, text, natural language, analysis

C2: information retrieval, analysis, document, summarization

C3: index, generate web album, web album generator

C4: computational linguistic, natural language generate, language generate

Practical experiments demonstrate that STC can be used to generate user profiles consisting of characteristic phrases. We are planning to perform some experiments with a predefined topic taxonomy, i.e. replace clustering by classification. It is also our objective to implement time window to account for profile „aging".

Most importantly, now we need to find a sufficient number of volunteers willing to participate in model web-browsing, in order to collect web pages corresponding to individual users' interests.

It is still a question of further research to implement additional applications such as search ranking (use profile to sort web pages returned by a search engine), query expansion, and web page pre-fetching based on user profile.

ACKNOWLEDGEMENTS

This work has been partly supported by grants No. MSM 235200005 and ME494.

REFERENCES

1.Diligenti M., Gori M., Maggini M.: A Unified Probabilistic Framework for Web Page Scoring Systems. In: IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 1, January 2004

2.Oyama S., Kokubo T., Ishida T.: Domain-Specific Web Search with Keyword Spices, In: IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 1, January 2004

3.Liu F., Yu C., Meng W.: Personalized Web Search for Improving Retrieval Effectiveness. In: IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 1, January 2004

4.Yu K., Schwaighofer A., Tresp V., Xu X., Kriegel H.P.: Probabilistic Memory-Based Collaborative Filtering, In: IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 1, January 2004

5.Maron M. E., Curry S., Thompson P.: An Inductive Search system: Theory, Design and Implementation. In: IEEE Transaction on Systems, Man and Cybernetics, vol. SMC-16, No. 1, January/February 1986, pp. 21-28.

6.Delgado J., Ishii N., Ura T.: Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents. In: Proceedings Second Int. Workshop, CIA 1998, pp. 206-215

7.Chen H.C., Chen A. L. P: A music recommendation system based on music data grouping and user interests. 2001. Retrieved from http://citeseer.org/ in February 2004.

8.Yimam D., Kobsa A.: Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. Retrived from http://citeseer.org/ in February 2004.

9.Mockus A., Herbsleb J. D.: Expertise browser: a quantitative approach to identifying expertise. In: Proceedings of the 24th international conference on Software engineering, Orlando, Florida, pp. 503-512, ISBN 1-58113-472-X, 2002

10. Wiener, P.: Linear pattern matching algorithms. In: Proceedings of the 14th IEEE Symposium on Switching and Automata Theory, pp. 1-11, 1973

11. McCreight E. M.: A space-economical suffix tree construction algorithm. In: Journal of the

ACM, 23:262-272, 1976

12. Zamir O., Etzioni O.: Web Document Clustering: A Feasible Demonstration; retrieved in February 2003 from CiteSeer: http://citeseer.org.