University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitní 8
30614 Pilsen
Czech Republic

User Profile Detection Based on Text Mining

State of the Art and Concept of Doctoral Thesis

Petr Grolmus

Technical Report No. DCSE/TR-2004-09
May, 2004

Distribution: public

Technical Report No. DCSE/TR-2004-09
May 2004

User Profile Detection Based on Text Mining

Petr Grolmus

Abstract

The old-fashioned search engine model based usually on Boole's algebra has become outdated. Current systems tend to meet users (customers) needs more precisely and hence they have to improve their basis. The searching process has to be based on deeper knowledge of individual user preferences – i.e. on user profiles.

This work presents an overview of recommender and expert finding systems. It also introduces the ProGen System rising at the University of West Bohemia. For the characteristic phrases selection from document collections is used the STC algorithm. Our system is able to work with multi-lingual document collections. A developed document language recognition is based on stop-word occurrence. We have developed a stemming algorithm combining both the dictionary-based and algorithm-based method. Finally, an outlook for the future work is sketched out.

This work has been partly supported the by Ministry of Education of the Czech Republic – grants No. MSM 235200005 and ME494.

Copies of this report are available on
http://www.kiv.zcu.cz/publications/
or by surface mail on request sent to the following address:

University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitní 8
30614 Pilsen
Czech Republic

Contents

1 Introduction

The great boom of the Internet started in 1990s. At the beginning of the 1990s there were only several hundred hosts in the Internet. According to the latest survey performed by the Internet System Consortium[1] there are now[2] almost 250 million hosts connected to the Internet (see Figure 1). The survey includes only hosts listed in DNS tables all over the world. The actual number of hosts in the Internet is undoubtedly greater. The expansion experienced since early 1990s implies increasing number of documents published by means of the fastest growing technology – the World Wide Web.

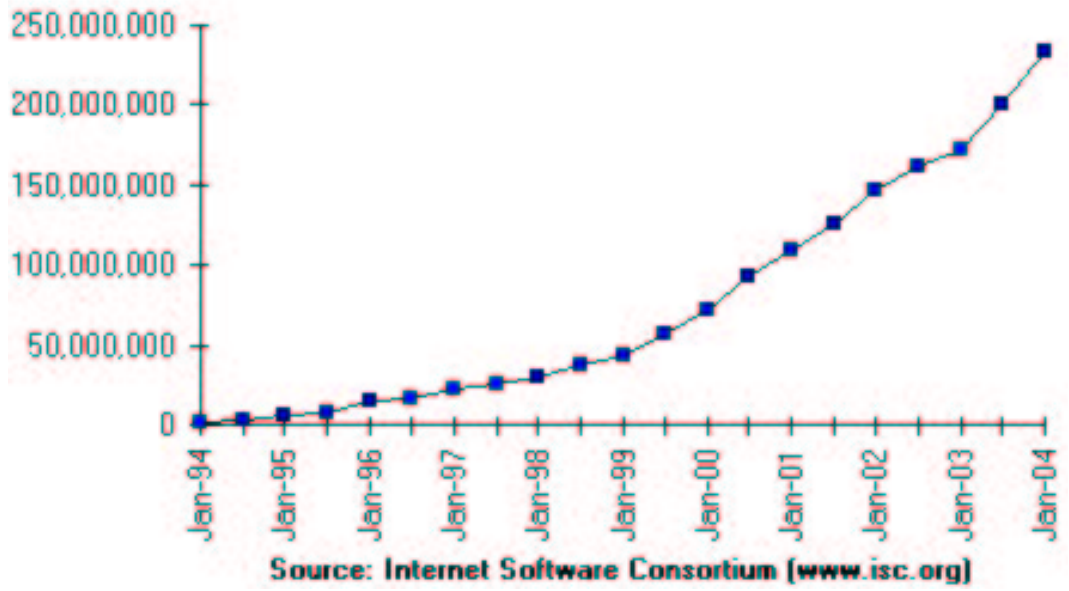

Figure 1: Measurable Growth of the Internet

New Internet users are at first amazed by possibilities offered to them by hyper-linked WWW documents. A first-time user starts reading one document and after several minutes (and clicks) finds himself reading another document focusing on a topic completely different from that of the first document in the chain. Later, the user's initial enthusiasm disappears. Mostly, after this disillusion, users get genuinely frightened of the number of accessible documents. This fact combined with the effort to minimize user frustration were the main reasons for the introduction of page-indexes (such as *Yahoo!*[3]) or *search engines* (also known as

[1] Internet System Consortium, Inc. – `http://www.isc.org/`

[2] April 2004

[3] `http://www.yahoo.com/`

search robots – for example AltaVista[4], WebCrawler[5] and more recently Google[6]) in mid-1990s.

However recently, the old-fashioned search engine model based usually on Boole's algebra has become outdated. In one's endeavor to survive in the field of searching in the Internet, tend search engines to allow a claim of theirs users/customers. They have to improve their basis – searching processes to offer more relevant documents on users' queries.

The ongoing "Internet battle" is the best time for the rise of recommender systems based on deeper knowledge of individual user preferences – i.e. on user profiles. Such systems should be able to meet users requirements precisely. But nothing comes free of charge. Recommender systems have usually longer response times. Most usually, they are working off-line. It is not uncommon for a recommender system to co-operate with a standard search engine when searching for suitable documents.

User profiles can also be used in expert finding systems. These are used to search for experts able to solve a given problem or to substitute an information source we do not have.

According to available literature, it seems that the definition of the term "recommender system" varies depending on the author. Some researchers use the concepts, "recommender system", "collaborative filtering" and "social filtering" interchangeably. On the other hand, others regard a "recommender system" as a generic descriptor that represents various recommendation/prediction techniques including collaborative, social, and content based-filtering, Bayesian networks and association rules.

[4] `http://www.altavista.com/`
[5] `http://www.webcrawler.com/`
[6] `http://www.google.com/`

2 Goals of the Research

This report focuses mainly on user profile detection and potential applications in various areas of human activities. Special attention is also paid to the ProGen (Profile Generator) System rising at the University of West Bohemia.

Chapter 3 contains a brief introduction to the classification of expert-finder and recommender systems as well as a description of existing solutions. Chapter 4 describes the ProGen System in detail. Chapter 5 gives a detailed description of the Suffix Tree Clustering algorithm used by the ProGen System to form clusters. Series of experiments have been carried out to show that our approach is feasible. Document collections used to test the system as well as the results of these experiments (and also of experiments carried out by real users) are listed in chapter 6. Actual state, usage alternatives, ongoing development process, and further research are all discussed in chapter 7.

Various research disciplines at universities and other large research institutions attract scientists of different specialty. Because of numerous locations of the university buildings combined with overlapping focus of faculties, it is often the 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 the similarity of their user profiles.

The main goal of our research is to develop user-profile generating system based on following WWW documents visited by users themselves. The focus is turn on multi-lingual environment considering users lingual abilities (many users are able to read documents in several languages). After many prolonged discussions, we have decided not to use information gathered from WWW proxy server logs. There are several reasons for this decision. First, in an organization equipped with a good connection to the Internet (typically universities), only few users are using proxy servers. Usually, they use a direct connection to the Internet. Second, in proxy-server logs, users are usually identified by IP addresses of computers they use. But there are places – such as student laboratories – where IP addresses do not belong to individual users. Lastly, we cannot rely on the difference between times recorded in consequent rows of the proxy-server log. These cannot be used to assess document relevance because we are not able to decide whether the user spent the time by reading the document or doing something else (reading mail, talking to colleagues, etc.).

Another aspect in which our system differs from common recommender systems is the fact that we do not require relevance feedback from the user. We suppose that the offering evaluation forms with recommended documents can annoy the user, who can easily start filling in random data. This behavior could be truly counter-productive. The only form of feedback we get from the user is the list recommended documents the user has visited.

We have developed a new process for stemming multi-lingual document collections. This stemming combines both dictionary and algorithm process methods. As the stemming process is highly language-dependent, we have also developed and implemented a technique for document language recognition based on stopwords occurrence. The language recognition was an essential step as a document collection of real users mostly has really different parameters than artificial testing collections. For example, real user collections usually consist of documents written in several languages. At the present time, we are able to recognize several languages including Czech, English, and German.

We want to use the ProGen Application to recommend interesting documents to users who have not visited them yet. We also want to use it to find domain experts, i.e. to build virtual communities. But, the usage of the system based on user profiles could be more wide. Another possible applications are for example query search expansion, search ranking with respect to user profile, web-page prefetching, etc.

This thesis is based on ProGen-related articles that have been already published, i.e. [9], [8], and [10].

3 General Concepts of User Profiling

The boom of "expert finder systems", or "recommender systems", started in 1980s and keeps drawing attention of many researchers in both commercial and academic world. There are two main reasons to use an expert finder system. First, we may want to find information or an answer to a specific question. In such a case an expert may serve as a substitute for another information source (such as a book, a scientific report, etc.). Second, we may want to find a suitable person to solve a given task – a consultant, an employee, a reviewer, a research worker, etc.

3.1 Division of Recommender Systems

Generally, recommender systems can be divided into two groups: *content-based* systems and systems based on *collaborative filtering*. The former search for similarities in documents visited or marked as interesting by a user and try to recommend other (non-visited) documents with similar content. Machine learning algorithms are used to classify documents as interesting or non-interesting (or relevant and non-relevant respectively). The latter do not offer recommendations based on user profile preferences but on similarities between the target user's profile and profiles belonging to other users registered in the system (the system recommends documents visited by other users with similar profiles). Collaborative filtering profiles are based on attributes specified explicitly by users. Both approaches are combined together in *hybrid recommender systems*.

We can also classify recommender systems as *internal* and *external* ones. This division is mainly based on where the expert finding takes place. In the case of internal systems we want to find an expert within our organization. On the other hand, external systems look for domain experts outside the organization (this approach is usually suitable for small organizations or companies).

3.2 Knowledge Profiles

A knowledge profile provides for suitable storage of the professional knowledge spectrum of a given user. Information systems differ in profile representation. Furthermore, an important difference usually lies in the method of user profiles harvesting. Commonly, these methods are based on attributes shown in tables 1 on page 8.

Systems based on knowledge profiles usually get results with higher precision than searching engines based on Boole's algebra.

Implicit attributes	Explicit attributes
• frequently visited intranet sections • documents authored by user • participation in newsgroups; his/her inquiries and responses in these newsgroups	• job description • professional interests • function and department • periodicals subscribed • participation in conferences

Table 1: Criteria of Expertise

3.3 History of Expert Finding Systems and Recommender Systems

According to Yimam [23], the origin of the expert finding systems dates back to mid-1980s. There are several systems worth mentioning:

HelpNet (Maron, et al. [14], 1986) – experimental system; it accepts requests for information and responds with names of source people ranked by computed probability that the individual provides satisfactory answer. This probability is computed using probabilistic models of information retrieval which combines the estimation of people's expertise in answering a question on the topic with the probability that a given user would be satisfied with the response provided by a source. Maron and his colleagues also envisioned such systems enabling the emergence of "a large, active and fruitful future network of informationally rich people providing help to one another".

Expert/Expert Locator (EEL) (Steeter and Lochbaum [20], 1988; Derr and Lochbaum [6], 1995) – also called *Bellcore Advisor* or *WhoKnows*; this system takes natural language requests for technical help or advice and, using Latent Semantic Indexing (LSI) method, returns pointers to "nearby" research groups. Using EEL, Berry and Dumais (1994) run a test by inputting people's description of their own research interests as query to compare how well the system managed to predict which of the 480 Bellcore work groups an expert belongs to. A web based version of this system is also described in Derr and Lochbaum (1995) that answers queries with documents by automatically generating hypertext links to mentioned names of experts in the organization.

GroupLens (Resnick, et al. [19], 1994) – is a system for collaborative filtering of netnews, to help people find articles they will like in the huge stream of available articles. News reader clients display predicted scores and make it easy for users to rate articles after they read them. Rating servers,

called Better Bit Bureaus, gather and disseminate the ratings. The rating servers predict scores based on the heuristic that people who agreed in the past will probably agree again. Users can protect their privacy by entering ratings under pseudonyms, without reducing the effectiveness of the score prediction. The entire architecture is open: alternative software for news clients and Better Bit Bureaus can be developed independently and can interoperate with the components they have developed.

ContactFinder (Krulwich and Burkey [12, 13], 1995) – it is an intelligent agent that monitors discussion groups and extract contacts in some specific areas which it then uses to respond to questions posted with referral to relevant contact. This system uses various heuristics both to extract contacts from email messages and to identify those postings that ask for technical help.

Answer Garden (Ackerman and McDonald [1],1996) – this system is basically question answering and routing system that answers questions for technical help by retrieving stored question and answer pairs, but also provides facilities to route un-answered questions to a defined group of experts. The primary way users locate answers in Answer Garden is by answering a branching series of multiple choice questions, starting at the control panel. Each question is presented in a structured text node. When the user selects node of the answers by clicking on the appropriate button, the next structured text node appears, and the user continues answering multiple choice branching questions until (s)he finds the question and answer for which (s)he is looking.

Referral Web (Kautz, et al. [11], 1997) – they suppose that the best way of finding an expert is through what is called "referral chaining" whereby a seeker finds the needed expert through referral by colleagues (social networks). It attempts to uncover existing social networks rather then provide a tool for creating new communities. The RefferalWeb is based on providing referrals by way of chains of named individuals. It builds its network model from public documents.

The Expertise Browser (Cohen, et al. [4], 1998) – enables people in organizations to use logged and indexed browse paths of experts to find relevant documents.

Memoir (Pikarakis, et al. [17], 1998) – the MEMOIR framework supports researchers working with a vast quantity of distributed information, by assisting them in finding both relevant documents and researchers with related interests. It is an open architecture based on the existing Web infrastructure. Key to the architecture is the use of proxies and the use of an open and extensible message protocol for communication: to support message

routing for dynamic reconfiguration and extension of the system, to collect information about the trail of documents that a user visits, and to insert links on-the-fly.

RAAP (Delgado, et al. [5], 1998) – a multi-agent system called RAAP (Research Assistant Agent Project) that intends to bring together the best of both worlds - *content-based* and *collaborative information filtering*. In RAAP, personal agents both helps the user (a researcher) to classify domain specific information found in the WWW, and also recommends these URLs to other researchers with similar interests. This eventually brings benefits for the users as they are recommended only peer-reviewed documents, especially if they perform information and knowledge intensive collaborative work, such as scientific research.

ER – Expertise Recommender (Ackerman and McDonald [2], 2000) – the system is composed of components that implement the functionality and heuristics they identified from their field study in a software development company. *Expertise Recommender* provides four major components called profiling supervisor, identification supervisor, selection supervisor, and interaction management that maintain and manipulate expertise profiles.

Expert Finder (Vivacqua and Lieberman, [21], 2000) – an agent that automatically classifies both novice and expert knowledge by autonomously analyzing documents created in the course of routine work. Expert Finder works in the domain of Java programming, where it relates a user's Java class usage to an independent domain model. User models are automatically generated that allow accurate matching of query to expert without either the novice or expert filling out skill questionnaires.

FieldWise (Fagrell, et al. [7], 2000) – a research project that has aimed at developing a knowledge management architecture for mobile work domains. The architecture developed, called FieldWise, was based on fieldwork in two organizations and feedback from users of prototype systems. FieldWise adds to the field of CSCW by offering an empirically grounded architecture with a set of novel features that have not been previously reported in the literature.

MRS – Music Recommendation System (Chen and Chen [3], 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 and Kobsa, [23, 24], 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 and Herbsleb, [16], 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.

Agentware Knowledge Server™ from Autonomy[7] – this commercial knowledge management system includes a feature that identifies an employees area of expertise based on the documents they access from and submit to the organizational Intranet.

AskMe Enterprise™ from AskMe Corporation[8] – is another solution from commerce sphere. It offers A comprehensive set of features that promote employee-to-employee knowledge and expertise sharing across the organization. These capabilities enable employees to engage in interactive "question & answer" sessions through various means, including email, instant messaging, portal, web browser or even a PDA.

[7] `http://www.autonomy.com/`
[8] `http://www.askmecorp.com/`

4 The ProGen System

This section describes the user profile generator named ProGen. It has been designed and partially developed at the University of West Bohemia in Pilsen.

4.1 System Design

Information concerning the behavior of a user is acquired filtering his (her) packets. The packet filter captures all user-generated requests for WWW documents – packets that are distributed via port 80 (standard HTTP port) – and stores them into a database for further processing (step 1 shown on Figure 2 on page 13). The most important advantage of the packet filter is the fact that it can be switched off at any time. Thus, it is possible to prevent undesirable alteration of the user profile by certain documents and it allows the user to maintain a certain level of privacy. The ProGen System users are identified by 32-byte-long randomly generated strings.

After collecting a sufficient number of packets (usually hundreds or thousands of URL requests), the system downloads these documents and generates an user document collection (step 2).

User profiles (interest clusters) are generated from document collections by means of the Suffix Tree Clustering (STC) algorithm, which is similar to creating an inverted list of phrases occurring in a document collection (steps 3 and 4 in Figure 2).

Upon forming clusters as described above, we can use them for variety of applications, e.g. co-operation with search engine (step 5) for the recommendation of interesting documents user has not visited yet (step 6). Other possible applications are for example finding domain experts, query search expansion, building virtual communities (collaborative filtering), search ranking with respect to user profile, etc.

4.2 Packet Filter

To use the system, any prospective user has to enroll using the system's WWW interface[9]. Immediately after enrolling, a software package containing the packet filter if offered for download. This software package also contains personalized configuration file with a randomly generated 32-character string – `IDENT`. Inside the system, the user is represented by this unpredictable string. Hence, it is not easy for other users to insert irrelevant URLs to influence user profile being gen-

[9] `http://profily.zcu.cz/`

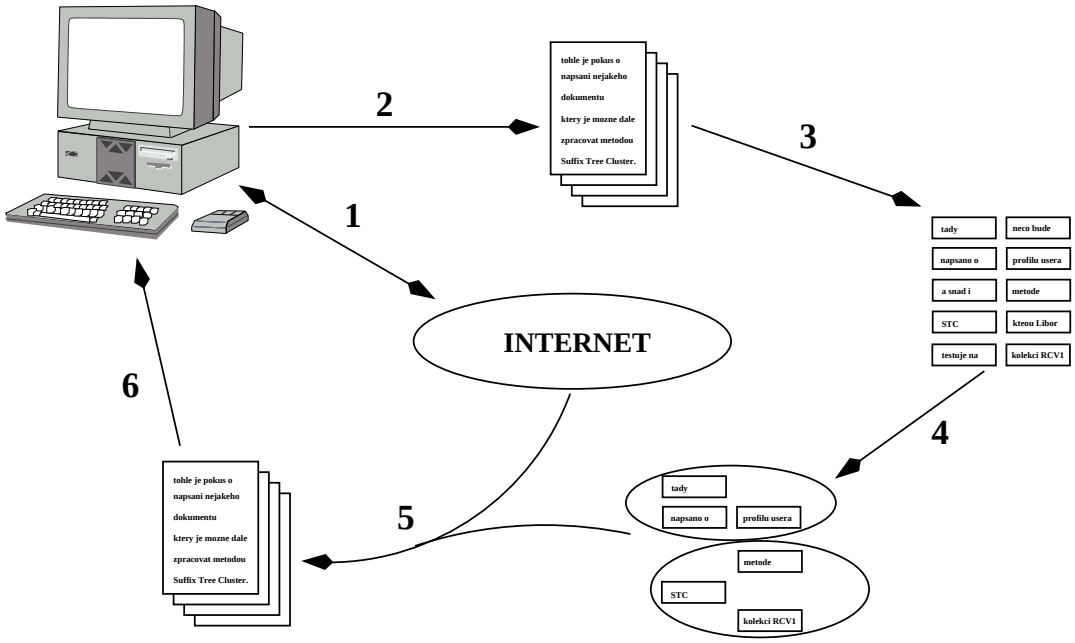

Figure 2: System design

```
<HOST>home.zcu.cz</HOST>
<USER>profil_insert</USER>
<DATABASE>web_profil_profil</DATABASE>
<PASSWORD>insert_only</PASSWORD>
<IDENT>4a613ce9f985b8ed85424ab4138304fc</IDENT>
```

Table 2: An Example of Packet Filter Configuration File

erated (clusters based on the user's interests). An example of a user configuration file is shown in Table 2.

4.3 Document Processing

4.3.1 Language Recognition

It is impossible to know the characteristics of a real user document collection. For example, we do not know the language used in individual documents in the collection. We cannot assume that the whole collection is written in one language only, such as English. Generally, users have different abilities and usually understand several languages. That is the reason why we have developed a document language identification method based on quantifying the occurrence of stop-words.

At the present time, we are able to recognize several languages including Czech, English, and German.

First of all, we generate a summary of all stop-words occurring in the document and then we compare the list of stop-words with groups of stop-word identifying individual languages. In case one of the languages contains (for example) more than 70% of all stop-words occurring in the document, we assume that the document is written in that language. On the other hand, if no language known to the system reaches the 70-% limit, the document is not submitted for further processing (to minimize errors). Language identification is fundamental for stop-word removal and stemming. There are homographic stop-words occurring in several languages (such as the Czech conjunction "a" and the English indefinite article "a"). Please note that it is much more difficult to distinguish between languages that are similar to each other – for example Czech and Slovak. Stop-word lists for languages mentioned above are described by Table 3 on page 15.

4.3.2 Stop-word and Punctation Removal

Once the document language is identified, we can remove all stop-words. The second step of preprocessing before stemming is the removal of punctation. These operations return a string of non-stemmed words.

4.3.3 Stemming

Stemming (also known as lemmatization) is the next step of the document cleaning process. Generally, we recognize two different stemming techniques: dictionary-based and suffix removal [18]. The latter is also known as the Porter's algorithm.

The dictionary method is based on a database containing derived words and their foundations. On the other hand, the Porter's algorithm tries to search for known suffixes inside the word being processed and to identify the longest suffix that may be possibly used in the language of the document. After removing the suffix, the relict has to contain at least 3 characters.

We are using a combination of both methods mentioned above for our purposes. The stemming function is illustrated in Figure 3 on page 15.

Characteristics of the stemming databases being used by the system are shown in Table 3 on page 15. Databases of derived words for Czech, English and German languages have been obtained from the *Ispell*[10] program distributed with the Linux operating system.

[10] `http://ficus-www.cs.ucla.edu/geoff/ispell.html`

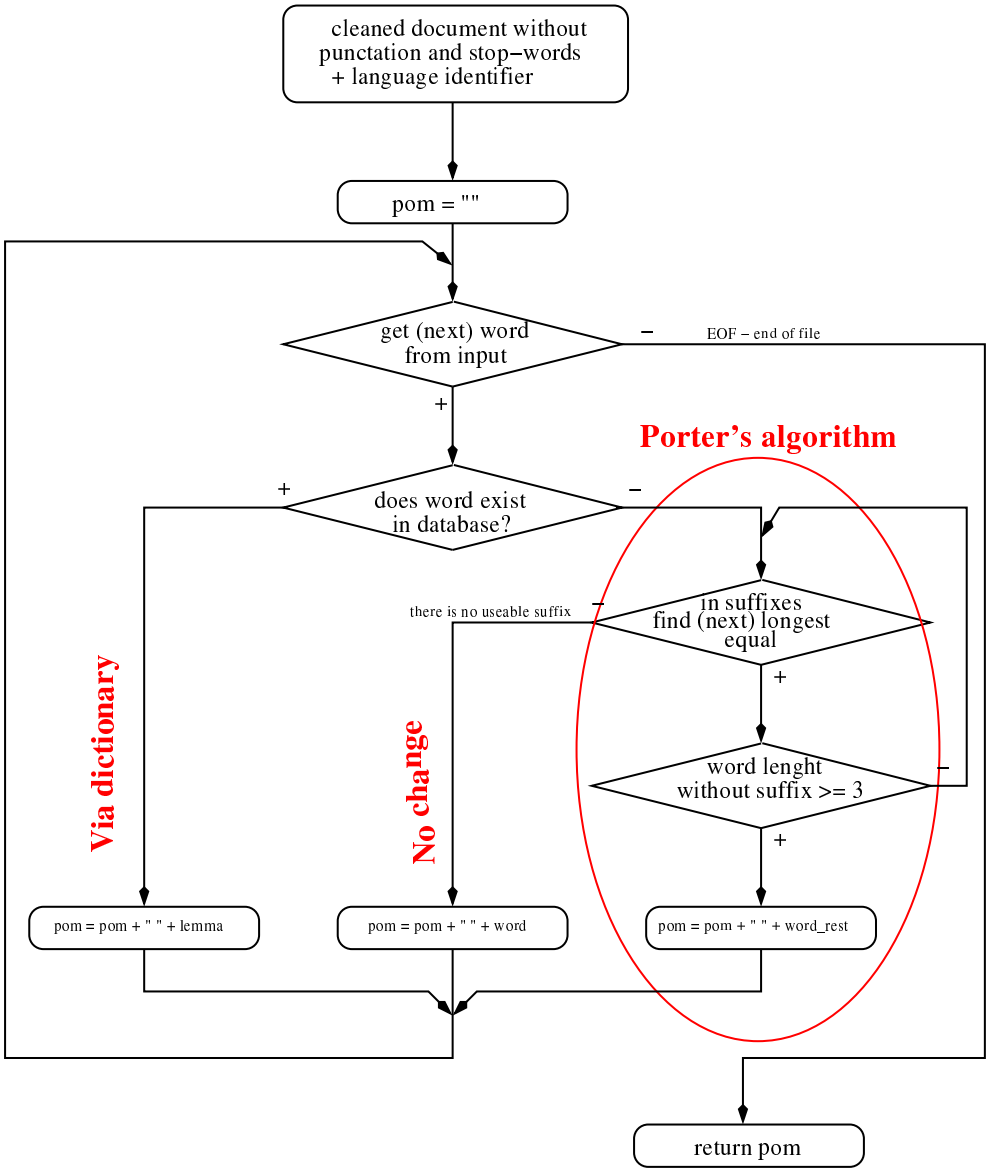

Figure 3: Document Stemming

	Czech	English	German
Words in dictionary	2 995 181	183 065	317 090
Stop-words	740	591	65
Suffixes	248	29	127

Table 3: Stemming database characteristics

Via dictionary	Porter's algorithm	No change
61.46 %	4.72 %	33.82 %

Table 4: The Utilization Rate of Individual Branches of the Stemming Function

```
<!DOCTYPE document_collection [
<!ELEMENT document_collection - - (description, document+) >
<!ELEMENT description - - (#PCDATA) >
<!ELEMENT document - - (document_id, url?, language?, title?, topiclist?,
                        keywords?, abstract? content) >
<!ELEMENT document_id - - (#PCDATA) >
<!ELEMENT url - - (#PCDATA) >
<!ELEMENT language - - (#PCDATA) >
<!ELEMENT title - - (#PCDATA) >
<!ELEMENT topiclist - - (topic+) >
<!ELEMENT topic - - (#PCDATA) >
<!ELEMENT keywords - - (#PCDATA) >
<!ELEMENT abstract - - (#PCDATA) >
<!ELEMENT content - - (#PCDATA) >
]>
```

Table 5: The XML Document Type Definition

We have tried to record the usage of individual branches of the stemming function. Results gathered while processing document collections obtained from testers are shown in Table 4 on page 16.

4.4 Document Representation

As soon as the user visits a sufficient number of WWW documents (visited URLs are stored in the database by a packet filter), the system downloads these documents for further processing. With regard to all our surveys in information retrieval, we had to define a standard representation for both real and testing document collections. It was reasonable to use XML. We have defined our own XML Document Type Definition (DTD) that fits our internal needs. The definition of the designed DTD is shown in Table 5 on page 16.

The whole document collection is included in the `document_collection` container. This container holds a set of document containers consisting of items describing each document, such as `document_id`, `language`, `title`, `topic`, `url`, and `content` of the document. Table 6 on page 17 shows a sample of the XML

```
<?xml version="1.0">
<!DOCTYPE document_collection system="http://profily.zcu.cz/dtd/dc.dtd">
<document_collection>
<document>
<document_id>1</document_id>
<language>english</language>
<content>cat eat cheese</content>
</document>
<document>
<document_id>2</document_id>
<language>english</language>
<content>mouse ate cheese too</content>
</document>
<document>
<document_id>3</document_id>
<language>english</language>
<content>cat ate mouse too</content>
</document>
</document_collection>
```

Table 6: Sample Document Collection from the STC Example

document collection based on the STC tree construction example described further in this document.

5 Suffix Tree Clustering

Suffix trees were published for the first time in 1970s in related works authored by Wiener[22] and McCreight[15]. It took many years for suffix trees to gain recognition and wider use. There were many reasons for this but one of the main ones probably was the fact that suffix trees were relatively greedy in their space requirements. Nowadays, the general opinion on suffix trees has changed because of our modern computer equipment (faster processing with sufficient amount of memory and disk space). Furthermore, we are working with relatively small collections of documents visited by users. Numbers of documents included in these collections do not exceed several thousand.

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 the user's web browser are passed to STC immediately after being processed by our stemming and stop-word filtering engines. Before we proceed to text stemming, it is necessary to identify document language, as stemming process is highly language-dependent.

Suppose that l is the maximum phrase length. We can create a suffix tree in the following manner:

1. Tree initialization (tree contains nothing but the root);
2. Let us read first l words $w_1 \dots w_l$ from input document (or less, if there are no more words);
3. Each input word w_i, where $i = 1, \dots l$, is inserted into the STC tree at level i so that the path from the root of the STC tree to the node inserted most recently leads through nodes associated with words $w_1 \dots w_{i-1}$. We have to save document numbers for each node (i.e. phrase) to keep track of 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 the following input document, continue with step 2

Looking at the 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, $\frac{s}{m}$ thereby being average number of words in document, r is a constant corresponding to proportional representation of the average document length with respect to the total number of words s. Constant $\frac{m}{k}$ being added to the results is to increase the number of characteristic phrases by one for each k documents in the collection.

The Suffix Tree Clustering algorithm is incremental. If we were able to store the representation of the tree we have built in a file, we could restore this tree from the file and append documents visited by the user since the last time the tree was processed. STC is also order independent, i.e. it does not depend on the document input order. Language independence is another feature of the STC algorithm. It means that if the user visits documents with the same topic in two different languages with similar frequency, he will probably get two similar clusters related to the subject, but each of them in different language.

5.1 Example of Characteristic Phrases Finding

There is a proverb that says: one picture is more than thousand words. Let us create a Suffix tree for the following three sentences (documents) and find out characteristic phrases. This example has been taken from [25].

Let us have documents:

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

The Suffix tree generated for documents listed above is shown in Figure 4 on page 20. The next step in the process is setting the appropriate weight for each node (phrase) in the Suffix tree. Suppose that phrase weight is computed as:

$$w_f = L_f \times N_f$$

where w_f represents the weight of the phrase f, L_f is its length in words and N_f is the number of documents containing the phrase f.

The final step is defining conditions for selecting characteristic phrases from all phrases. For example, for conditions defined as

$$(w_f \geq 3) \wedge (N_f \geq 2)$$

we obtain characteristic phrases: *cat ate*, *ate*, *ate cheese* (see Figure 4).

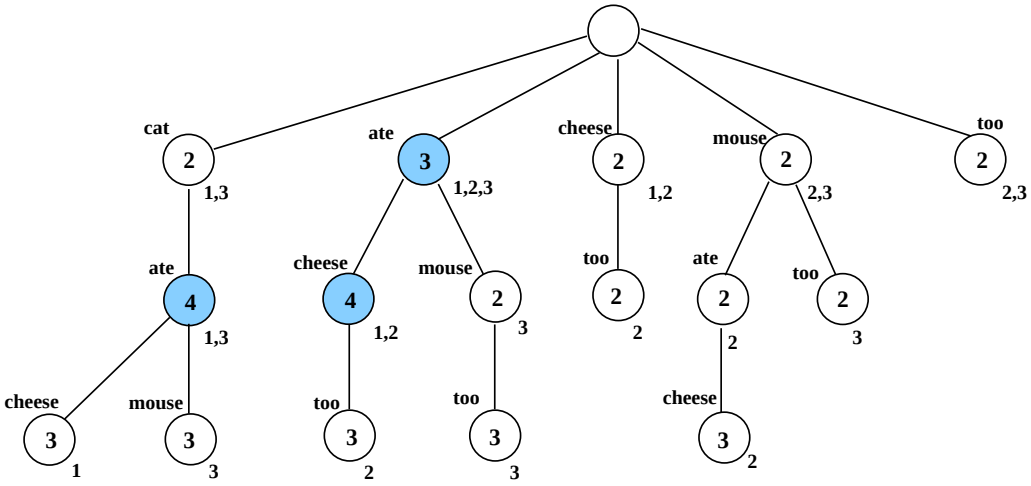

Figure 4: Suffix tree example

5.2 Cluster generation

In order to determine the discriminatory power of phrases stored in document collection, the weight factor is computed for each phrase:

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

where $L(f_i)$ represents the length of a particular phrase (expressed in significant terms), $N(f_i)$ is the number of occurrences of such a phrase, tf_{ij} is the number of occurrences of phrase f_i in document d_j, m is the total number of documents in the collection, and df_i is the number of documents containing phrase f_i . By using the square power of $L(f_i)$ we accentuate the importance of long phrases occurring less frequently.

We are computing the similarity of characteristic phrases identified by the 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"). A similarity between two different phrases is identified on the basis of a set of documents containing both phrases:

$$\left(\frac{|D_m \cap D_n|}{|D_m|} \geq \phi \right) \wedge \left(\frac{|D_m \cap D_n|}{|D_n|} \geq \phi \right)$$

where D_m and D_n represent documents containing phrases f_m and f_n, respectively. ϕ 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 has to be selected (and vice versa).

The higher value of ϕ threshold is, the fewer edges between phrases are formed, decreasing the average size of profile clusters, while increasing the number of these clusters. We have not specified any upper limit for the number of clusters yet, although it is reasonable to assume that one user will not be characterized by more than three or four identifiable areas of interest.

Alternatively, we can measure phrase-to-phrase similarity by means of *support* and *confidence*:

$$s(X \Rightarrow Y) : \frac{X \cap Y}{n} = P(X \cup Y)$$

$$c(X \Rightarrow Y) : \frac{X \cap Y}{X} = P(Y \mid X)$$

respectively

$$s(D_m \Rightarrow D_n) : \frac{|D_m \cap D_n|}{|D|}$$

$$c(D_m \Rightarrow D_n) : \frac{|D_m \cap D_n|}{|D_m|}$$

where D_m and D_n represent documents containing phrases f_m and f_n, respectively. D represents the quantity of all documents in the input document collection.

6 Document Collections and System Results

6.1 Testing Collections

Before collecting sufficient amount of testing data, we have tested the system using various document collections, such as Reuters Corpus Volume One - RCV1 (more than 800 thousand documents), 20Newsgroups (20 thousand), and ČTK - Czech Press Agency (131 thousand). The precision of our recommender system varies from 85% to 98% depending on the input parameters. An overview of the testing collections is shown in Table 7 on page 22.

	ČTK	**RCV1**	**20Newsgroups**
Number of documents	130 955	806 791	19 997
Number of words	29×10^6	193×10^6	5.23×10^6
Average document length	159,0	88,4	155
Shortest document (in words)	10	10	1
Longest document (in words)	5 721	3 996	6 695
Average number of topics a document is classified to	1,7	3,2	1.0
Number of topics	42	103	20

Table 7: Testing Collection Overview

We have concluded a successful negotiation with the Czech Press Agency, which has supplied us with a document collection containing the Agency's archive of news published in 1999. It was a great success for us, as it provided us with the only usable Czech document collection suitable for testing purposes. Testing the process on different languages was one of the goals of our work.

All testing collections are classified and documents are attributed to one or more topics. Classification of the Reuters Corpus Volume One is hierarchical (tree structure). ČTK and 20Newsgroups have flat structure (non-hierarchical).

Top-level classes for RCV1 and ČTK:

ČTK: Politics, Sport, Companies, Police & Law

RCV1: Corporate/Industrial, Economics, Government/Social, Markets

20Newsgroups collection documents are distributed evenly (1,000 documents in each of the 20 classes).

6.2 Experimental Results

The experimental phase is conducted as follows: a specific testing collection category, such as sports or economics, is subject to testing by randomly selecting approximately 2/3 of documents included in this category. These training documents represent a group of visited documents; user's interest respectively. Then, the system searches for frequently occurring phrases, selects characteristic phrases, and defines applicable profile clusters using the STC algorithm. The remaining 1/3 of documents belonging to the current category (relevant documents) is mixed with a certain number of documents from other categories (irrelevant documents).

In this step, we are trying to find profile-cluster phrases in these testing documents. For searching purposes, we create a variable vector of characteristic phrases (a phrase-centroid) representing each profile cluster. Accomplishing that, we can compute cosine-metric similarity between a testing document D and characteristic phrases of profile cluster C_i ($i = 1 \ldots$ number of clusters found):

$$Sim(C_i, D) = \frac{\sum_1^H (w_h \times d_h)}{\sqrt{\sum_1^H (w_h)^2 \times \sum_1^H (d_h)^2}},$$

where H is the number of characteristic phrases for cluster C_i, w_h is the weight of h^{th} phrase, and d_h is the number of occurrences of the h^{th} phrase in document D. If $Sim(Ci, D) \geq \psi$, then $D \subset Ci$, where ψ is a specific threshold, which means that document D is considered relevant for the user.

Knowing that the document actually belongs or does not belong to the category, we are able to measure precision (P) and recall (R) of the designed process as follows:

$$P = \frac{S_R}{S_R + S_N}, \qquad R = \frac{S_R}{S_R + N_R},$$

where S_R represents the set of selected relevant documents, S_N selected irrelevant, and N_R not selected relevant documents.

Experimental results for the Czech document collection (ČTK) are shown in Table 8 on page 24 as well as in Figure 5 on page 24. Clusters of these experiments are generated from 6,362 single-topic documents (politics). Precision and recall measurements are computed using 3,181 relevant documents (classified with the same topic) and the same number of irrelevant documents (others topics).

Results concerning the English document collection RCV1 are shown in Table 9 on page 25 and in Figure 6 on page 25. Experiments have been carried out under

Threshold ψ	0.5	0.6	0.7	0.8	0.85	0.9	0.95
Precision	57.47	61.55	65.47	85.22	85.28	89.99	90.34
Recall	97.60	94.60	82.80	63.61	54.01	44.35	32.69

Table 8: Experimental Results of the ČTK Collection Evaluation

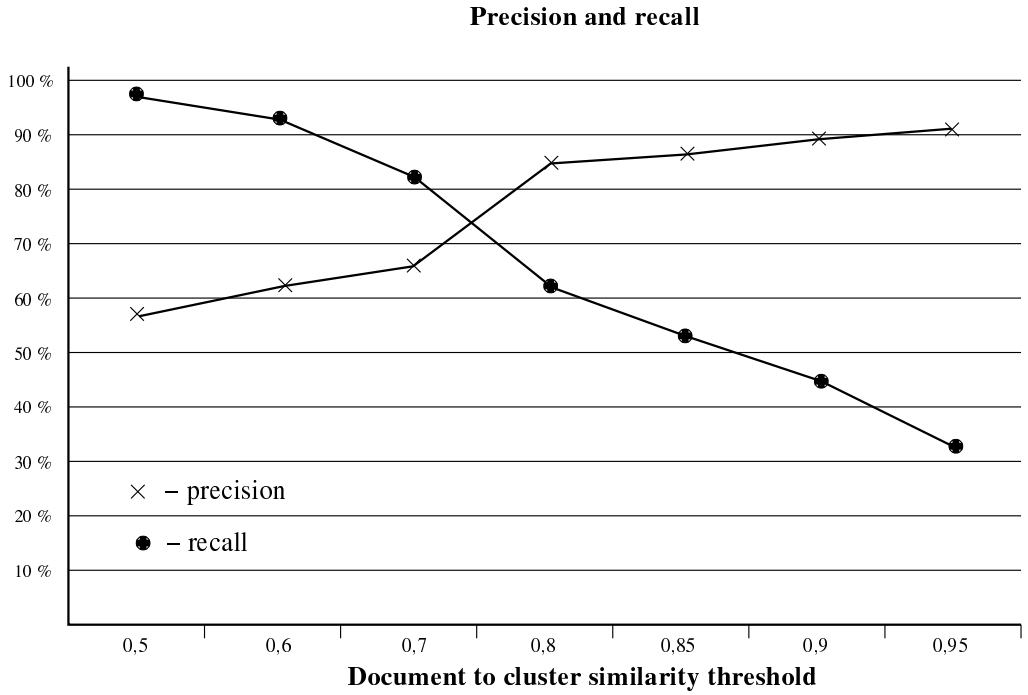

Figure 5: Precision and Recall in the Czech ČTK Collection

conditions similar to the previous case. The only difference is using documents classified with four topics instead of one topic. The results of both experiments (Czech and English document collections) are compared in Figure 7 on page 26.

Clusters generated from the ČTK news reports (translated from Czech into English for the purpose of this article):

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

Threshold ψ	0.5	0.6	0.7	0.8	0.85	0.9	0.95
Precision	57.09	62.61	72.08	83.35	88.41	92.39	93.00
Recall	83.54	75.05	55.59	42.75	33.18	27.02	19.25

Table 9: Experimental Results of the RCV1 Collection Evaluation

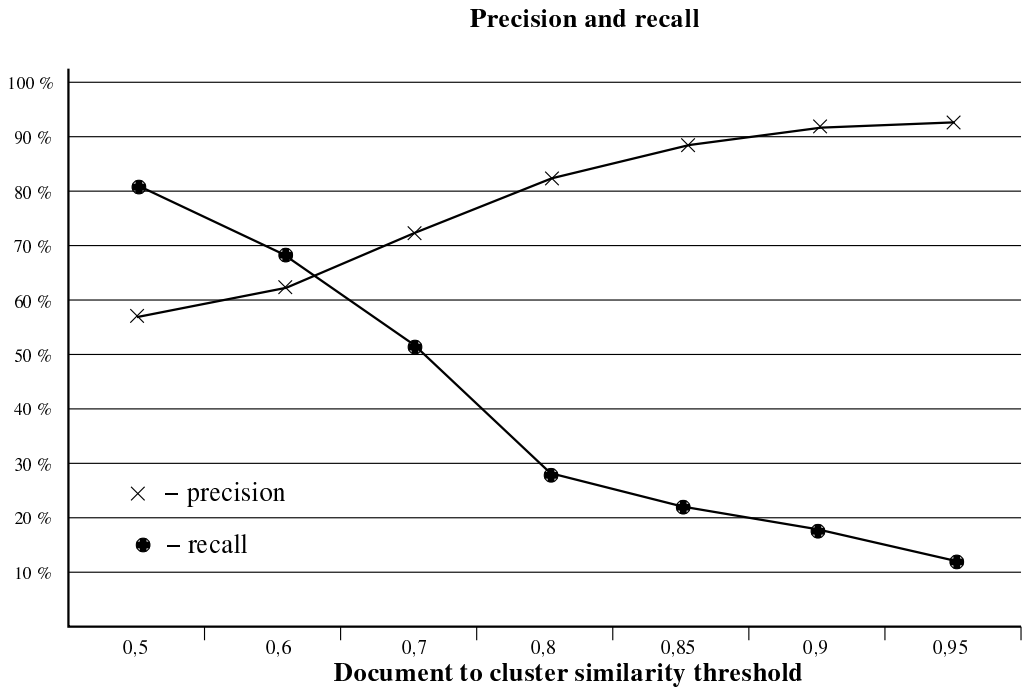

Figure 6: Precision and Recall in the English Collection RCV1

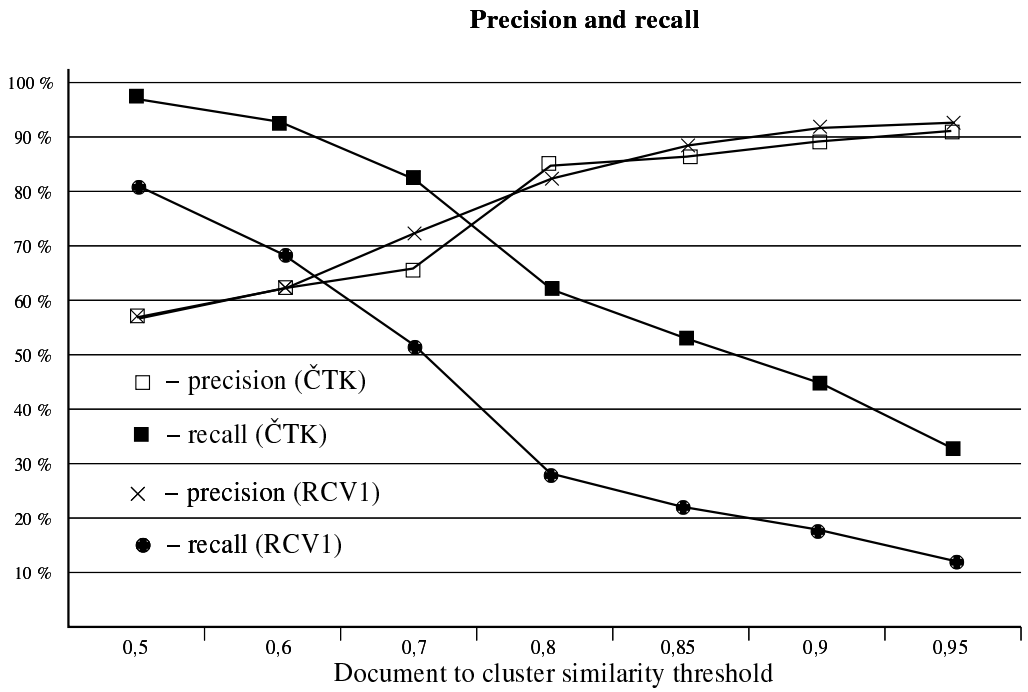

Figure 7: The Comparison of Czech and English Document Collection Results

Number of documents:	6 362
Number of words(without stop-words):	1 003 072
STC nodes:	1 397 413
Time consumed:	approximately 10 minutes

Table 10: Cluster Generation Characteristics

Cluster generation characteristics mentioned above are shown in Table 10 on page 26.

6.3 Real Environment Testing

In the course of last few months, we have used testers to check the provisional system. Generally, these testers are PhD. students seeking information related to their specialization. The testing group has both advantages and disadvantages. The main disadvantage is the number of its members (less then 10). The main advantage is their narrow specialization leading to higher precision in document recommendations. On the other hand, the narrow specialization can also be considered a disadvantage due to the low diversity in generated clusters.

An example of a cluster generated by the system for a real user is shown hereinafter:

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

Clusters are built from a collection of documents visited by a user interested in document summarization. The third cluster presents the user's private interest in web-based galleries publishing photographs.

7 Conclusion and Further Research

Practical experiments demonstrate that STC can be used to generate user profiles consisting of characteristic phrases and is suitable enough for relatively small collection of documents visited by individual users.

To sum up, our approach differs from similar systems published so far (see References on page 30) in many ways. A packet filtering application is used to collect WWW documents visited by users. User profile generation is based on characteristic phrases located in STC trees. Characteristic phrases are selected from STC trees by a modified TF×IDF method. Our system is able to work with multi-lingual document collections. Document language recognition is based on stop-word occurrence. We have developed a stemming algorithm combining both the dictionary-based and algorithm-based method.

We do not use information stored in WWW proxy server logs, as it does not seem suitable for the purpose at hand. Given good Internet connectivity, few users use WWW proxy services. Furthermore, users can use different IP addresses over time (typically students in computer laboratories). Finally, we cannot fully rely on time intervals separating individual log records while assessing relevance, because we can never be sure if the user has spent all the time reading a document or doing something else (reading mail, discussion with colleagues, etc.).

Major features of the ProGen System are already set but minor ones are still changing in the time while the system is being developed. The main goal of the ProGen System, appointed to recommend interesting (relevant) documents for individual users, is not implemented yet. The co-operation with the search engine (e.g. *Google*) for searching documents (recommendation adepts) is still missing. We suppose that we can use for example the first 50 documents returned from the search engine and classify them as members of clusters of the user profile (which means the user interest taxonomy in such a case) by means of classification methods.

The important fact is that user profiles are adaptable. User preferences may vary over time. In order to make a profile "live" with user, the system must accept changes in the user's interests. This is also one of our future goals – user profile adaptability. User profile adaptability should be ensured by regenerating profiles using documents visited within a specific period only (a "time window"). Among other parameters, page viewing frequency and link visit percentage can be used for this purpose.

If we were able to store the STC tree of a given user in a file, we would minimize the amount of time needed to regenerate user profiles as proposed in the previous paragraph. We would have to complement information stored in the STC tree by the timestamp of the last visit for each document used for the STC tree creation.

In the time of following STC tree regeneration, we would simply restore the last state of the tree from the file and remove documents visited outside the "time window" (documents visited before a given date respectively). Documents visited by the user since the last STC tree regeneration are included in the reduced STC tree. We assume that rebuilding the STC tree from the file can reduce the processing time considerably. Plans for the future also include the development of an application generating artificial document collections. They could be either document selections from real document collections (e.g. RCV1 or ČTK) by specific attributes (document length, word combinations, etc) or entirely artificial (built from combinations of unmeaning words – e.g. *word1*...*word1000*). Experiments on such collections can confirm or confute the presumption that document recommendation based on user profiles generated from multi-word phrases gives more precise results than single-words. Due to the absence of user feedback (in our opinion it could annoy the user) we can assess the precision and recall of real-user document collections by means of these experiments only. Experiments on these artificial collections could also lead to improving the accuracy when measuring the parameters of the ProGen System (as well as its precision and recall).

We intend to perform various experiments with a predefined topic taxonomy, i.e. replace clustering with classification so that we can create virtual communities (users with similar profiles come under the same topic of this taxonomy). We could also combine the ProGen System with other works of Text-Mining Research Group at the University of West Bohemia. For example – it could be interesting to combine ProGen with the automated text document summarization survey, as it would allow us to recommend not only pure URL addresses of relevant documents but also their generated summary.

Most importantly, 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 user interests.

References

[1] Ackerman, M. S., and McDonald, D. W. – Answer Garden 2: Merging Organizational Memory with Collaborative Help. In: Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'96), Novemeber, 1996, pp. 97-105, 1996

[2] Ackerman, M. S., and McDonald, D. W. – Expertise Recommender: A Flexible Recommendation System and Architecture. In: Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work (CSCW'00), Philadelphia, PA, pp. 231–240, 2000

[3] Chen, H.-C., and Chen, A. L. P – A music recommendation system based on music data grouping and user interests. Recieved from `http://citeseer.org/` in February 2004.

[4] Cohen, A. L., Maglio, P. P., and Barrett, R. – The Expertise Browser: How to Leverage Distributed Organizational Knowledge. Paper presented at the Workshop on collaborative Information Seeking at CSCW'98; Seatle, WA, 1998

[5] Delgado, J., Ishii, N., and Ura, T. – Content-based Collaborative Information Filtering: Actively Learning to Classifify and Recommend Documents. In: Proceedings Second Int. Workshop, CIA'98, pp. 206–215, 1998

[6] Derr, M. A., and Lochbaum, K. E. – A Web Application for Finding Potential Collaborators. A paper presented in W3C Workshop on WWW and Collaboration, Cambridge, MA, pp. 11–12, 1995

[7] Fagrell, H., Forsberg, K., Sanneblad, J. – FieldWise: A Mobile Knowledge Management Architecture. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work, 2000

[8] Grolmus P., Hynek J., and Ježek K. – Frequent Phrase Mining in Text to Generate User Profiles. In: Proceedings of ITAT'03, R.Lences (Ed.), 2003 (in Czech)

[9] Grolmus P., Hynek J., and Ježek K. – User Profile Identification Based on Text Mining. In: Proceedings of ISIM'03, Miroslav Beneš (Ed.), Marq, s.r.o, Ostrava (2003), pp. 109–116. (ISBN 80-85988-84-4)

[10] Grolmus P., Hynek J., and Ježek K. – A Web-Based User-Profile Generator: Foundation for a Recommender and Expert Finding System. Accepted article for proceedings of ElPub 2004 (already sent), Brasilia, 2004

[11] Kautz H., Selman B., and Shah M. – Referral Web: combining social networks and collaborative filtering. In: Communications of the ACM, Volume 40, Issue 3, pp. 63–65, ISSN:0001-0782, 1997

[12] Krulwich, B., and Burkey, C. – ContactFinder: Extracting indications of expertise and answering questions with referrals. In: The working Notes of the 1995 Fall Symposium on Intelligent Knowledge Navigation and retrieval. Technical Report FS-95-03, The AAAI Press, pp. 85-91., 1995

[13] Krulwich, B., and Burkey, C. – The ContactFinder Agent: Answering bulletin board questions with referrals. In: Proceedings of the 1996 National Conference on Artificial Intelligence (AAAI-96), vol. 1, pp. 10-15, 1996

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

[15] McCreight, E. M. – A space-economical Suffix Tree Construction Algorithm. Journal of the ACM, 23:262–272, 1976

[16] 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

[17] Pikarakis A., et al, – MEMOIR: Software Agents for Finding Similar Users by Trails. In: Proceedings of the Third International Conference on the Practical Applications of Intelligent Agents and multi-Agent Technology (PAAM-98), London, UK, pp. 453–466, 1998

[18] Pokorný, J., Snášel, V., Húsek, D. – Dokumentografické informační systémy, Karolinum, Prague 1998 (ISBN 80-7184-764-X)

[19] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. – GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, Chapel Hill, NC: pp. 175-186, 1994

[20] Steeter, L. A., and Lochbaum, K. E. – An Expert/Expert Locating System based on Automatic Representation of Semantic Structure. In: Proceedings of the Fourth IEEE Conference on Artificial Intelligence Applications, Computer Society of the IEEE, San Diego, CA, pp. 345-349.

[21] Vivacqua, A., Lieberman, H. – Agents to assist in finding help. In: Proceedings of Conference on Human Factors in Computing Systems, pp. 65–72, ISBN:1-58113-216-6, 2000

[22] Wiener, P. – Linear pattern matching algorithms. In proceedings of the 14^{th} IEEE Symposium on Switching and Automata Theory, pp. 1–11, 1973

[23] Yimam D. – Expert Finding Systems for Organizations: Domain Analysis and the DEMOIR Approach. In: ECSCW 99 Workshop – Beyond Knowledge Management: Managing Expertise, 1999

[24] 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.

[25] Zamir, O., Etzioni O. – Web Document Clustering: A Feasible Demonstration; retrieved from CiteSeer – `http://citeseer.org/`, in February 2003

Glossary

ČTK – stand for Česká tisková kancelář (Czech Press Agency); the provider of our Czech testing document collection.

DTD – stands for Document Type Definition; an SGML definition for a markup language.

HTML – stands for HyperText Markup Language. Markup language for description of WWW pages; based on SGML.

HTTP – stands for Hypertext Transfer Protocol; a protocol communication for between client's WWW browser and WWW server.

ISO – Organization for International Standards, its specification respectively.

Lemmatization – see *stemming*.

Packet – a data unit used to communication between at least two hosts in the network.

Packet Filtering – a technique for selecting packets by the given pattern.

Precision – an information retrieval performance measure that quantifies the fraction of retrieved documents which are known to be relevant.

RCV1 – stands for Reuters Corpus Volume One; an English testing document collection provided by Reuters agency.

Recall – an information retrieval performance measure that quantifies the fraction of known relevant documents which were effectively retrieved.

SGML – stands for Standard Generalized Markup Language; metalanguage for electronic documents, standard specification ISO 8879.

STC – stands for Suffix Tree Clustering; a developed technique for the document clustering from Suffix Trees.

Stemming (also known as *lemmatization*) – a technique for reducing words to their grammatical roots.

Stop-words – words which occur frequently in the text of a document. Examples of stop-words are articles, prepositions, and conjunctions.

URL – stands for Uniform Resource Locator; it is used to address any Internet resource, including Web pages.

User Profile – is a model of user's interests created (or learned) by the recommender system. It is e.g. a set of keywords extracted from documents visited by user.

WWW – stands for World Wide Web; fastest growing technology of the Internet.

XML – stands for eXtensible Markup Language; a subset of SGML, defined for the Web. In XML it is easier to define new markup languages.

Publications of the Author

1. Grolmus, P., Ježek, K., Bokr, J. – Poiskovye uslugy zapadoczeskavo universiteta v Plzene. Vestnik Sankt-Peterburskavo universiteta technologii i disajna, No.7, 2002, ISSN 1029-8606, Russia, 2002

2. Grolmus, P., Hynek, J., Ježek, K. – User Profile Identification Based on Text Mining. In proceedings of ISIM'03, Mir. Beneš (Ed.), pp. 109–116, ISBN 80-85988-84-4, Czech Republic, 2003

3. Grolmus, P., Hynek, J., Ježek, K. – Frequent Phrase Mining in Text to Generate User Profiles. In proceedings of ITAT'03, R. Lences (Ed.), Slovakia, 2003

4. Grolmus, P., Hynek, J., Ježek, K. – A Web-Based User-Profile Generator: Foundation for a Recommender and Expert Finding System. Accepted paper for proceedings of ElPub 2004, Brasilia, 2004

Technical Reports of the Author

1. Grolmus, P. – Návrh a realizace vyhledávací služby na ZČU. Annual report, 34 pages, 2002

2. Grolmus, P. – Automatické generování uživatelského profilu. Annual report, 30 pages, 2003

Publications Related on Users' Profiles, Expert Finder Systems, and Recommender Systems[11]

1. Balabanović, M. – Learning to Surf: Multiagent Systems fro Adaptive Recommendation. A dissertation at Stanford University, 1998

2. Broder, A. J. – Data Mining, the Internet, and Privacy. In proceedings of International WEBKDD'99 Workshop, pp. 56–73

3. Chakrabarti, S. – Mining the Web. Morgan Kauffmann Publishers, ISBN 1-55860-754-4, 2003

4. Chan, P. K. – Constructing Web User Profiles: A Non-Invasive Learning Approach. In proceedings of International WEBKDD'99 Workshop, pp. 39–55

5. Chen, C. C., Chen, M. C., Sun, Y. – PVA: A Self-Adaptive Personal View Agent System. Retrieved from `http://citeseer.org/` in June 2003

6. Cooley, R., Mobasher, B., Srivastava, J. – Web Mining: Information and Pattern Discovery on the World Wide Web. Retrieved from `http://citeseer.org/` in June 2003

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

8. Fu, Y., Sandhu, K., Shih, M. – A Generalization-Based Approach to Clustering of Web Usage Sessions. In proceedings of International WEBKDD'99 Workshop, pp. 21–38

9. Haas, S. W., Grams, E. S. – Page and Link Classifications: Connecting Diverse Resources. Retrieved from `http://citeseer.org/` in May 2002

10. Han, E.-H., et al. – WebACE: A Web Agent for Document Categorization and Exploration. Retrieved from `http://citeseer.org/` in May 2002

11. Herlocker, J. L., Konstan, J. A., Terveen, L.G., Riedl, J.T. – Evaluating Collaborative Filtering Recommender System. ACM Transactions on Information Systems (TOIS), Volume 22 Issue 1, January 2004

[11]Non-mentioned publications in the main report (but related to the line of study); alphabeticaly ordered

12. Kao, H.-Y., Lin, S.-H., Ho, J.-M., Chen, M.-S. – Mining Web Informative Structures and Content Based on Entropy Analysis. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No.1, January 2004

13. Kleinberg, J. M. – Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, Vol. 46, No.5, September 1999

14. Koval, R., Návrat, P. – Intelligent Support for Information Retrieval in the WWW Environment. In proceedings of Advances in Databases and Information Systems – 6^{th} East European Conference, Slovakia, 2002

15. Lawrence, S., Giles, C. L., Bollacker, K. – Digital Libraries and Antonomous Citation Indexing. IEEE Computer, Vol. 32, No.6, 1999

16. Li, W.-S., Candan, K. S., Vu, Q., Agrawal, D. – Query Relaxation by Structure and Semantics for Retrieval of Logical Web Documents. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No.4, July/August 2002

17. Lian, W., Cheung, D. W. – An Efficient and Scalable Algorithm for Clustering XML Documents by Structure. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No.1, January 2004

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

19. Masand, B., Spiliopoulou, M. – Web Usage Analysis and User Profiling. Springer, 1999, ISBN 3-540-67818-2

20. Mladenic, D. – Text-Learning and Related Intelligent Agents: A Survey. IEEE Intelligent Systems, July/August 1999

21. Mobasher, B., Dai, H., Luo, T., Sun., Y., Zhu, J. – Integrating Web Usage and Content Mining for more Effective Personalization. In proceedings of ECWeb, 2000

22. Murray, D., Durrell, K. – Inferring Demographic Attributes of Anonymous Internet Users. In proceedings of International WEBKDD'99 Workshop, pp. 7–20

23. Oyama, S., Kobuko, T., Ishida, T. – Domain-Specific Web Search with Keyword Spices. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No.1, January 2004

24. Polčicová, G., Návrat, P. – Semantic Similarity in Content-Based Filtering. In proceedings of Advances in Databases and Information Systems – 6^{th} East European Conference, Slovakia, 2002

25. Ricardo B.-Y., Berthier, R.-N. – Modern Information Retrieval. Addison-Wesley, ACM Press, ISBN 0-201-39829-X, 1999

26. Rijsbergen van C. J. – Information Retrieval. Butterworths, London, 1979

27. Yu, H., Han, J., Chanf, K. C.-C. – PEBL: Web Page Classification without Negative Examples. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No.1, January 2004

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