IDENTIFYING SIMILAR USERS BY THEIR SCIENTIFIC PUBLICATIONS TO REDUCE COLD START IN RECOMMENDER SYSTEMS

Stanley Loh [1,2], Fabiana Lorenzi [2,3], Roger Granada [1]
Daniel Lichtnow [1,3], Leandro Krug Wives [3] and José Palazzo Moreira de Oliveira[3]
[1]*UCPEL Universidade Católica de Pelotas, Rua Felix da Cunha 412, Pelotas, RS, Brasil*
[2]*ULBRA Universidade Luterana do Brasil, Av. Farroupilha, 8001, Canoas, RS, Brasil*
[3]*UFRGS Universidade Federal do Rio Grande do Sul, Av.Bento Gonçalves 9500-Bl IV,Porto Alegre-RS,Brasil*

Keywords: User profile, User profile similarity, Collaborative recommender systems.

Abstract: This paper presents investigations on representing user's profiles with information extracted from their scientific publications. The work assumes that scientific papers written by users can be used to represent user's interest or expertise and that these representations can be used to find similar users. The goal is to support similarity evaluations between users in a model-based collaborative recommender. Representing users by their publications can help minimizing the *new user* problem. The idea is to avoid the necessity of asking users to evaluate a set of items or give some information about their preferences, for example. In scientific communities, particularly on digital libraries and systems focused on the retrieval of scientific papers, this is an interesting feature. We have conducted some experiments to compare different techniques to represent the papers (*title*, *keywords*, *abstract* and *complete text*) and two kinds of text indexes: *terms* and *concepts*. Furthermore, two distinct similarity functions (Jaccard and a Fuzzy function) were applied on these representations and then compared with the goal of finding similar users.

1 INTRODUCTION

Collaborative filtering (or social information filtering) is one of the most used techniques in recommender systems. There are two kinds of collaborative filtering techniques: item-item and user-user. Item-item based identifies correlations between items in order to define new items to be recommended to users; recommended items are those similar to items already associated to the user. User-user based technique evaluates the similarity between users to find users with similar tastes or needs; in this case, items to be recommended are those associated to similar users.

In the user-user technique, two approaches may be applied: memory-based and model-based. In the memory-based one, similarity between users is evaluated by identifying items with common ratings in historical data from two users. In the model-based approach, items associated to users are employed to define a model for each user; after that, the similarity between users is evaluated by identifying the similarity between their models (Wang et al., 2006).

The approach presented in this paper is based on models of users, instead of using memory-based techniques. Memory-based approach has the advantage of being less complex (less parameters have to be tuned). In contrast, model-based approaches generate compact models and suffer less with the sparsity problem (Wang et al., 2006)).

Collaborative filtering suffers from some problems such as the **cold start** or **startup problem**, the **sparsity problem**, and the **shilling problem**. The cold start or startup problem happens when there are few ratings for an item or made by a user so the system has no sufficient data to give recommendations; the latter case is also called the *new user* problem. The sparsity problem happens when there are few common items rated by users, and the shilling problem when someone tries to favor some particular items (Adomavicius and Tuzhilin, 2005).

In universities and research groups it is very common to allocate new members since new members (usually students) arrive frequently. In this case, a recommender system will not have any register about the preferences, ratings and interactions of these new users. Thus, it is necessary to collect some infor-

Loh S., Lorenzi F., Granada R., Lichtnow D., Krug Wives L. and Palazzo Moreira de Oliveira J.
IDENTIFYING SIMILAR USERS BY THEIR SCIENTIFIC PUBLICATIONS TO REDUCE COLD START IN RECOMMENDER SYSTEMS.
DOI: 10.5220/0001823405890596
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

mation before producing recommendations or, alternatively, produce bad recommendations and improve the results of future interactions by using users' feedbacks. Then, users have to expect some time for receiving recommendations, that is, until the system elaborates a profile (a model of his/her interest) or until the user rate some items.

The focus of this paper is the *new user* problem in recommender systems targeted on learning environments. Despite of the existing differences between recommendation to consumer (related to goods) and recommendation to learners systems, as pointed out by (Drachsler et al., 2008), the same problem is found. This paper minimizes this problem by analyzing user's publications, generating recommendations for new learners and researchers.

The goal of this paper is to compare different techniques to generate a user's model analyzing his/her publications and identifying his/her area of interest. Similar users are then identified by comparing the users' models. By similar users, we mean users that have interest in common scientific areas with similar proportion.

The techniques used here are able to compare different sections of the papers (e.g., *title*, *keywords*, *abstract* and *full text*) and two kinds of text indexes: *terms* and *concepts*. Furthermore, two distinct similarity functions (Jaccard and a Fuzzy function) were applied on the representations to find similar users.

The results give some hints that can be used to improve existing collaborative filtering systems that will be able to elaborate an initial profile or the user model, thus minimizing the *new user* problem.

The paper is structured as follows. Section 2 presents some papers related to model-based collaborative filtering and to the representation of users' interest and expertise. Section 3 details the proposed method for representing users' profiles with different techniques. Section 4 presents the experiments carried out for evaluating the proposed techniques and discusses the results. Section 5 presents a scenario to illustrate the application of method in a recommender system. Finally section 6 summarizes the contributions and discusses future works.

2 RELATED WORK

The *new user* problem is a relevant one, and is usually minimized by the use of techniques to identify the similarity between users. In (Adomavicius and Tuzhilin, 2005), for example, a model-based approach is used for the analysis of users' similarity. In their case, techniques based on clustering and on Bayesian networks are employed to find similar users. Similarly, (Stoilova et al., 2005) propose evaluating the similarity between users through their bookmarks analysis. Other particular way to evaluate similarity between users is to examine their social relations or networks, as performed in (Spertus et al., 2005).

The problem is that there is a lack of information about users, and it results in low quality recommendations. Some initiatives try to solve this problem. For instance, in the Movielens System, new users are asked to rate some movies when they start using the system in order to create an initial profile (Rashid et al., 2002). The problem is that, sometimes, a new user does not have time or willing to do this initial rating.

An alternative way to the evaluation of the similarity between users and to minimize the new user problem is to analyze scientific publications of these users. This is important in the context of recommending items from a digital library or information sources in learning situations. In this context, publications can be used under different approaches. In an item-item approach, collaborative filtering systems can consider similarity between papers in the following way: papers written by users are similar to papers cited inside these papers, as employed by (McNee et al., 2002). In a user-user memory-based approach, systems can evaluate the similarity between users by analyzing common publications or common vehicles where papers were published.

In these approaches, one important question is how to analyze users' publications and represent their content. Under a user-user model-based approach, systems can generate a model for each user from the texts associated to him/her (written, read or cited) and compare these models to infer the similarity between users. This approach is employed by (Middleton et al., 2003). The limitation is that profiles are generated analyzing papers browsed by users, what is far from minimizing new user problem.

In (Dumais and Nielsen, 1992), different techniques for representing the expertise or interest of a conference reviewer are presented. In this case, they utilize family names, keywords and abstracts extracted from papers supplied by the reviewers as the best representatives of their knowledge. Similarly, (Yarowsky and Florian, 1999) use a centroid (a term vector) generated from papers representatives of the reviewer expertise. (Basu et al., 2001) represents reviewers with information extracted from papers that are written by the reviewers or referenced by them in their home pages (titles, abstracts and keywords are used to represent papers). These studies do not compare the use of titles, keywords and abstracts among them (the context of the paper is query reformulation). Other limitation is that the cited work does not evaluate the use of the whole text as representative of the user interest. Analyzing the presented related work, it is possible to notice the need for comparing different extracts from publications (scientific papers) in order to represent expertise or interest areas of people. Our goal in the current paper is to evaluate different representations of texts as representatives of users profiles, such as parts of the text (titles, keywords, abstracts and full text) and different kinds of indexes (terms versus concepts). Representations are extracted from texts of scientific papers written by the users. The final goal is to apply these profiles in collaborative systems to find similar users.

3 INVESTIGATION

The goal of this paper is to investigate techniques for finding similar users in the context of user-user model-based collaborative filtering systems or methods. We propose the utilization of users' publications (scientific papers written by them) to represent their interest. We assume that publications of the users are already collected and separated by title, keywords, abstract and complete text. The publications may be collected by analyzing the curriculum vitae of the user or in public sources such as Scholar Google (http://scholar.google.com), Citeseer (http://citeseer.ist.psu.edu), and the Brazilian BDBComp (http://www.lbd.dcc.ufmg.br/bdbcomp).

This work compares parts of the papers (titles, keywords, abstracts and complete texts) for representing profiles in order to find similar users. Furthermore, the work examines what kind of index better represents texts in this context. We tested *terms* versus *concepts*; *terms* are single words extracted from texts following the full-text indexing method (after eliminating stopwords); *concepts* are extracted from probabilistic analysis of words presented in the text and correspond to nodes of a domain ontology.

In addition, we compare two functions for calculating similarity between texts: the Jaccard method and a Fuzzy equation proposed by (Loh et al., 1998). The results of this investigation must show what kind of technique is better suited for finding similar users. In order to achieve these goals, we have defined a general process for analyzing texts with the following steps:

1. **Tokenization:** separating single words from each text;
2. **Stopwords elimination:** terms like prepositions and articles should be disregarded;
3. **Identification of the relative frequency:** for each token (term): relative frequency is the frequency of the term in the text divided by the total number of terms in the text;
4. **Creation of a weighted term vector:** term and relative frequency for representing each text.

In the next subsections, we explain the different techniques used to extract information from user's publications and to represent the user model.

3.1 Text Parts: Title vs. Abstracts vs. Keywords vs. Complete Text

One of the investigations of this paper concerns the structure of the scientific papers used in the user profile to represent the user interest. Following the work of (Basu et al., 2001), we selected the *titles*, the *keywords*, the *abstracts* and the *complete texts* as paper representatives.

The goal is to know if simple parts (as titles and keywords) can achieve better results for representing user interest and for finding similar users. If so, we would not need to process bigger parts of the text as abstracts or the full text of the paper.

3.2 Text Indexes: Terms vs. Concepts

Another investigation is related to the kind of text representations (indexes). The majority of current studies use term vectors to represent texts. However, terms (especially single words) are prone to problems due to the use of synonyms (different words for the same meaning), polysemy (the same word with many meanings) and lemmas (words with the same radical, like the verb "to marry" and the noun "marriage") (Chen, 1994).

One alternative approach that has been used with success is the use of concepts instead of terms to represent texts. Concepts have been used also in Information Retrieval in order to index and retrieve documents. As pointed by (Lin and Chen, 1996), the concept-based retrieval capability has been considered as an effective complement to the prevailing keyword search or user browsing. Concepts belong to the extra-linguistic knowledge about the world (Sowa, 2000). They are expressed by words but in fact they represent *things* in a higher level (entities and events of the reality). Concepts are identified in texts with the help of a domain ontology. A domain ontology is a description of *things* that exist or can exist in a domain (Sowa, 2000) and it contains the vocabulary related to the domain (Guarino, 1998).

In the presented work, the ontology is implemented as a set of concepts in a hierarchical structure (a root node, parent-nodes and child-nodes). Each concept has associated to it a list of terms and their respective weights. Weights are used to state the relative importance or the probability of the term for identifying the concept in a text and they are defined by a traditional supervised learning process (like a Bayesian one). The relation between concepts and terms is many-to-many, that is, a term may be presented in more than one concept and a concept may be described by many terms.

The ontology is used to identify themes in texts using a probabilistic method that compares the terms presented in the text and the terms associated to the concept. A threshold is used to determine if the concept is presented or not in the text. The procedure is similar to the one presented in (Loh et al., 1998). Therefore, the investigation intends to compare users profiles composed by terms or by concepts. Terms and concepts are extracted from textual parts of the papers (titles, keywords, abstracts or complete papers).

3.3 Similarity Function: Jaccard vs. Fuzzy

There are different similarity functions for comparing texts. Cosine and Euclidean Distance are two of the most usual. The former evaluates the cosine of the angle formed by two vectors representing texts in a Cartesian plan. The latter calculates the distance between the two vectors in a Cartesian plan. One limitation of these functions is that they evaluate common attributes but fail to compute attributes that do not appear in the vectors (Willet, 1998).

Jaccard coefficient (Equation 1) is used to measure similarity between sets. It is defined as the size of the intersection divided by the size of the union of 2 sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \qquad (1)$$

In our case, it can be used to take in account attributes that do not appear in one of the vectors. Thus, the similarity degree between two vectors is calculated by the number of common attributes divided by the total number of attributes without counting repetitions (number of attributes in the first vector plus number of attributes in the second vector minus number of common attributes). However, the Jaccard coefficient fails to compute weighted vectors, that is, the weights associated to the attributes in the vectors are not utilized in the calculation. This can bring some misleading especially when dealing with texts. For example, terms that appear with different frequencies in different vectors will lead to an equal similarity.

For this reason, we use a different similarity function that regards the weights of the common attributes and also computes non-common attributes. This function was presented by (Loh, 2001). As shown in equation 2, the degree of similarity between two texts (vectors) is calculated by the sum of the degrees of equality of the common attributes weights divided by the total number of attributes found in both vectors.

$$gs(X,Y) = \frac{\sum_k^{h=1} gi_h(a,b)}{n} \qquad (2)$$

where: gs is the degree of similarity between texts X and Y; h is an index for the terms that are common to X and Y; k is the number of terms that are common to X and Y; n is the total number of terms in both documents (not counting repetitions); gi is the equality degree between weights of the term h in each vector (weight a in X and weight b in Y).

The equality degree between the weights is measured by equation 3 and it follows the work presented in (Pedrycz, 1993).

$$gi(a,b) = \frac{1}{2}\left[(a \to b) \wedge (b \to a) + (\overline{a} \to \overline{b}) \wedge (\overline{b} \to \overline{a})\right] \qquad (3)$$

where: $\overline{x} = 1 - x$; $a \to b = [c \in [0,1] \mid a \times c \leq b]$; and $\wedge = min$.

The equation takes into account the fact that an attribute may have different degrees of importance in different texts. Instead of calculating the average or the product between two degrees, the function determines the degree of equality between them. For example: if an attribute h (that is common to both texts being analyzed) has a weight of 0.9 in one text and 0.3 in the other, the average would be 0.6, equal as if the weights were 0.6 in both texts. In the same sense, the product of weights 0.9 and 0.4 would generate a result equal to two weights 0.6. However, weights 0.6 are more similar to each other than 0.9 to 0.3 or 0.9 to 0.4.

The experiments were performed using the similarity functions on vectors with different kinds of text attributes. Attributes may be a term presented in the text or a concept identified in the text as described early.

4 EXPERIMENTS AND EVALUATIONS

Experiments were undertaken in order to validate the methods described in the previous section. We selected 12 authors with scientific papers published in important conferences held in Brazil. These authors were grouped in pairs according to the area where they usually publish. Six areas were defined: Database, Software Engineering, Computers in Education, Artificial Intelligence, Computer Networks and Neural Networks.

In the next step, for each selected author, we collected 3 recent papers (in Portuguese) written by the author. The papers were collected from the Brazilian Digital Library of Computer Science (http://www.lbd.dcc.ufmg.br/bdbcomp/) and from the Brazilian Academic Google (http://scholar.google.com.br). Each author is represented by his/her papers according to the different techniques compared in this paper (explained in section 3).

For the experiment, each part of the paper (*titles*, *keywords*, *abstracts* or *complete text*) were used to represent the user interest. After, *terms* and *concepts*, with the respective weights, were extracted from the above texts (representations).

For the experiments involving concepts, a domain ontology for Computer Science was employed. The ontology was created based on the ACM classification for Computer Science. The high level concepts are similar to those of the ACM in the first level but we created more detailed levels (subdivisions of areas) to express more specific knowledge. However, the child concepts are quite different, resulting in a different hierarchy of concepts (or areas).

In order to define the terms and weights associated to each concept, a supervised learning process was conducted. Training texts, selected by experts, were analyzed by the TFIDF method (Salton and McGill, 1983) to generate the terms and weights. After, experts reviewed the ontology adding word variations with the same weight as the principal. A normalization step was applied over the weights to avoid a great variation in the limits from one concept to other.

In each experiment, the goal was to evaluate the similarity among authors, using the different techniques for representing the author's interest. A matrix of similarity between authors was then generated and pairs of authors were formed associating to each author the one among the 12 that was most similar to him/her. To compare the performance of each technique, we evaluated if the correct pair for each author was found by the techniques. That means that a total of 12 evaluations were performed for each technique. The percent of correct assignments were used as measure.

Table 1 shows the results got from *titles* and *keywords* and table 2 shows the results got with *abstracts* and *complete texts*. We have analyzed *concepts* and *terms* for each part of the paper (titles, keywords, abstracts and complete texts). Each column represents the percent of pairs correctly assigned by each technique or condition. We have run the experiments with two similarity functions: Jaccard and Fuzzy Function. The first row shows the performance results got from Jaccard function and the second row shows the results of the Fuzzy function.

Table 1: Results of the experiments: titles and keywords.

	Titles		Keywords	
Function	Concept	Term	Concept	Term
Jaccard	16.6%	0%	16.6%	66.6%
Fuzzy	16.6%	0%	25%	66.6%

As we can see in table 1, when using *terms*, *keywords* achieved a better performance (66.6% of correct assignments) with both similarity functions. In contrast, when using *concepts*, *abstracts* achieved better performance (58.3% with the Fuzzy function). However, with the Jaccard similarity, *abstracts* had the same performance (41.6%) as *complete texts*. This result is assumed as normal since the similarity function influences the performance as will be discussed in the next sections.

Table 2: Results of the experiments: abstracts and complete texts.

	Abstracts		Complete Texts	
Function	Concept	Term	Concept	Term
Jaccard	41.6%	8.3%	41.6%	41.6%
Fuzzy	58.3%	0%	50%	41.6%

In the same sense, we can say that the choice of the representative paper (*title*, *keywords*, *abstract* or *complete text*) is influenced by the kind of index employed (*terms* or *concepts*). However, the results suggest that using *keywords* with *terms* is better; this performance (66.6%) is 14% better than the second best performance (58.3% with *abstracts* and *concepts* using the Fuzzy function).

Analyzing the use of *terms* versus *concepts*, we can see that using Jaccard function, *concepts* performed better than *terms* with *titles* and *abstracts*. Jaccard lost in performance with *Keywords* and it had equal performance with *Complete Texts*. However, using the best performance with Jaccard was due to *terms* with *keywords* (66.6%).

Using the Fuzzy function, *concepts* performed better than *terms* with *titles*, *abstracts* and *Complete Texts* but lost with *keywords*. However, the best performance with Fuzzy function was due to *terms* (over *keywords*) with 66.6% of correct assignments. It is also interesting to note that *terms* achieved, in all experiments, the best performance (66.6% with keywords) and the worst performance (no hit with *titles* and only 8.3% with *abstracts* and Jaccard function).

The results confirm that *concepts* are more appropriate to be used with longer texts that represent papers (as for example, *abstracts* and *complete texts*). When the number of words is too small (as in *titles* and *keywords*), the performance with *concepts* is far from good. Another conclusion is that *terms* are more appropriated to be used with *keywords* and that this combination (*terms* and *keywords*) is the best one.

Comparing Jaccard versus Fuzzy similarity function, in 3 of the 8 conditions, the Fuzzy similarity function achieved a better performance than the Jaccard function, losing 1 case and tied in 4. Using concepts, the Fuzzy function performed better in 3 paper representatives (*keywords*, *abstracts* and *complete texts*) and tied in one (*titles*). Using *terms*, Jaccard achieved a better performance with *abstracts* and tied in 3 paper representatives (*titles*, *keywords* and *complete texts*). However, this win with *abstracts* was with a precision of only 8.3%.

This analysis leads us to conclude that the Fuzzy function performs better than Jaccard and it can be employed in whatever situation. The reason may be that it is important to regard the weight of the attributes as the Fuzzy function does and as the Jaccard does not.

5 APPLICATION SCENARIO

This section shows a simplified scenario of recommendation. The objective of this scenario is just illustrating the use of the proposal approach in a recommender system. The Figure 1 shows an overview of architecture. The architecture consists of 3 modules.

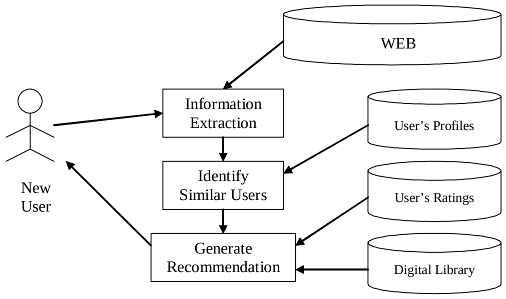

Figure 1: Application Scenario.

1. The Information Extraction Module receives an *id* of a new user (name, email, for example) The necessary *id* depending on repository. Using this *id* the module extracts information about user's publication from Web (e.g. Scholar Google,DBLP);
2. The Identify Similar User Module receives information about publication (a set of terms) and retrieves information about old users (profiles contains a set of keywords). The module calculates the similarity among users using the similarity functions described on section 3.3. The most similar users are identified and this information is sent to Recommendation Module;
3. The Recommendation Module retrieves information about similar users, generates and sends the recommendation to the relevant user. The recommendation is a set of items that similar users have given good rates. Good rates means items that users have used in the past (in this case the evaluation is implicit), or items that old users have given good rates in an explicit way.

Without identifying similar users, it will be necessary to generate recommendation using only terms related to the user (like in a search engine) or ask the user to evaluate some items (to build an initial profile).

6 CONCLUDING REMARKS

The paper presented investigations on different techniques for representing user profiles for similarity evaluation in user-user model-based collaborative recommenders. The work assumes that scientific papers written by users can be used to compose the user profile, representing the user interest or expertise.

Techniques were created to compare different parts of the papers (*title*, *keywords*, *abstract* and *complete text*) to be used as their representatives. Other techniques were used to compare two kinds of text indexes: *terms* and *concepts*. Furthermore, two distinct similarity functions (Jaccard and a Fuzzy function) were applied on the representations to find similar users.

Our evaluations show that the best performance is achieved with the combination of *terms* and *keywords* (in both similarity functions). It is important to say that the choice of the paper representative is influenced by the kind of index used. In the future, it is necessary to use a bigger sample sets in the experiments and others similarity measures can be tested (cosine, for example). However some preliminary conclusions rose after the experiments were:

1. If using *terms* instead of *concepts* for indexing texts, prefer to select *keywords* as paper representatives;
2. If using *concepts*, prefer to select *abstracts* as paper representatives;

3. The Fuzzy function is not suited to be used with the combination *abstracts* + *terms*, but in all other cases it outperforms the Jaccard similarity;
4. If needing to use *title*, *abstracts* or *complete texts* as paper representatives, prefer to use *concepts* as text indexes;
5. If using *keywords*, prefer to use *terms*; and
6. It is not necessary to use *complete texts* as paper representative; *complete texts* do not give the best performance and have additional burden of processing.

The final suggestion is to use the Fuzzy function with the combination of *terms* to index *keywords* extracted from papers. One of the reasons may be that authors select keywords that better represent the content of the papers and human decisions are still the best choice. However, it is interesting to note that even titles did not perform well, leading to the supposition that titles are not good representatives of the content of the papers or that authors fail in choosing words for titles. The result is a little surprising since we initially expected that complete texts would have the best performance. However, this finding is similar to the one presented by (Brutlag and Meek, 2000) that e-mail headers perform so well as message bodies for classifying e-mail messages, with the additional advantage of reducing the number of features to be analyzed. One possible reason for this surprising finding is that complete texts allow identifying many themes while titles and keywords concentrate in less and more specific themes.

The method for identifying themes in texts consider many possibilities and this can mislead the similarity evaluation, since many non-common themes can appear when comparing two authors. In this sense, (Kraft et al., 2006) found out that the number ideal of terms used in a query, in a search engine system, should be between 5 and 9 what show that a concept can be represented by a small set of terms. Another supposition is that increasing the threshold for considering themes in texts may bring less and more specific themes.

This is a point for a future work. For now, we can only say that *complete texts* have the most normal performance comparing *concepts* versus *terms* or Jaccard versus the Fuzzy function. In all the other 3 paper representatives (titles, keywords and abstracts), the difference between the best and the worst performance was too great.

In the same way, we noted that the best performance (66.6%) is still far from the desired one. This limitation can be due to the discussed before or due to the number of publications used for each user (only 3). Future works must evaluate the number of papers sufficient for representing the user's interest. However, we preview that, if the author publish papers in many different areas, the result will not be better. Thus, maybe to use a bigger number of documents are not going to produce better results.

It is important to notice that is necessary to find out areas of interest with a small number of documents. Some users do not have a lot of documents. In this sense, documents with less co-authors and documents where user is the first author should represent better users interest. Besides it is important to consider too that a great number of terms and documents are going to compromise the system's performance. In this sense, there are works related to document clustering and document classification where the use of a limited number of terms is proposed (Koller and Sahami, 1997), (Chang and Hsu, 2005).

Other possible cause of the bad performance may be the lack of advanced methods for term processing as stemming or n-grams. A future investigation must evaluate if mistakes can be corrected using one of these methods.

In the case of *concepts*, we do not associate the bad performance to the domain ontology. The ontology used in the experiments was evaluated in other works for classifying scientific papers and achieved results close to 90% of accuracy.

We are conducting an experiment to analyze the curriculum vitae of authors in order to discover his/her interest areas along the time and infer sequential patterns on changes of interest. This is very important point, because in general, persons with similar interests must be persons with similar interests at the same time (or almost). There are some examples of papers related to temporal effects on the performance of the recommender systems (Ding and Li, 2005).

We should remember that the results of this work can be applied to minimize the *new user* problem in a model-based collaborative recommender, through the use of a different kind of characteristic to represent the user's interest. Using the user's scientific publications, the similarity between users can be evaluated without the user having to rate items. Besides, the methods can be used to identify persons with similar profiles. A future work consists on the application of the techniques in a real recommender system to reproduce the scenario of section 5.

ACKNOWLEDGEMENTS

This work is partially supported by CNPq, Conselho Nacional de Desenvolvimento Científico e

Tecnológico, Brazil and CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brazil.

REFERENCES

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

Basu, C., Hirsh, H., and Cohen, W. (2001). A study in combining multiple information sources. *Journal of the Artificial Intelligence Research (JAIR)*, 14:231–252.

Brutlag, J. and Meek, C. (2000). Challenges of the email domain for text classification. In *7th International Conference on Machine Learning (ICML 2000)*, pages 103–110, Stanford University, USA.

Chang, H.-C. and Hsu, C.-C. (2005). Using topic keyword clusters for automatic document clustering. *IEICE - Trans. Inf. Syst.*, E88-D(8):1852–1860.

Chen, H. (1994). The vocabulary problem in collaboration. *IEEE Computer*, 27(5):2–10.

Ding, Y. and Li, X. (2005). Time weight collaborative filtering. In *CIKM ’05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 485–492, New York, NY, USA. ACM.

Drachsler, H., Hummel, H. G. K., and Koper, R. (2008). Personal recommender systems for learners in lifelong learning networks&#58; the requirements, techniques and model. *Int. J. Learn. Technol.*, 3(4):404–423.

Dumais, S. T. and Nielsen, J. (1992). Automating the assignment of submitted manuscripts to reviewers. In *15th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 233–244, Copenhagen, Denmark.

Guarino, N. (1998). Formal ontology and information systems. In *International Conference on Formal Ontologies in Information Systems - FOIS’98*, pages 3–15, Trento, Italy.

Koller, D. and Sahami, M. (1997). Hierarchically classifying documents using very few words. In *ICML ’97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 170–178, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kraft, R., Chang, C. C., Maghoul, F., and Kumar, R. (2006). Searching with context. In *WWW ’06: Proceedings of the 15th international conference on World Wide Web*, pages 477–486, New York, NY, USA. ACM.

Lin, C.-h. and Chen, H. (1996). An automatic indexing and neural network approach to concept retrieval and classification of multilingual (chinese-english) documents. *IEEE Transactions on Systems, Man and Cybernetics*, 26(1):1–14.

Loh, S. (2001). *Concept-based approach for knowledge discovery in texts (in Portuguese)*. PhD thesis, Federal University of Rio Grande do Sul.

Loh, S., Wives, L. K., and Oliveira, J. P. M. (1998). Concept-based knowledge discovery in texts extracted from the web. *ACM SIGKDD Explorations*, 2(1):29–39.

McNee, S., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A., and Riedl, J. (2002). On the recommending of citations for research paperss. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, pages 116–125.

Middleton, S. E., Shadbolt, N. R., and Roure, D. C. D. (2003). Capturing interest through inference and visualization: ontological user profiling in recommender systems. In *International Conference on Knowledge Capture KCAP03*, pages 62–69, New York. ACM Press.

Pedrycz, W. (1993). Fuzzy neural networks and neurocomputations. *Fuzzy Sets and Systems*, 56(1):1–28.

Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., and Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. In *IUI ’02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134, New York, NY, USA. ACM.

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Sowa, J. F. (2000). *Knowledge representation: logical, philosophical, and computational foundations*. Brooks/Cole Publishing Co, Pacific Grove, CA.

Spertus, E., Sahami, M., and Buyukkokten, O. (2005). Evaluating similarity measures: a large-scale study in the orkut social network. In *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery and data mining KDD 05*, pages 678–684.

Stoilova, L., Holloway, T., Markines, B., Maguitman, A. G., and Menczer, F. (2005). Givealink: mining a semantic network of bookmarks for web search and recommendation. In *Proceedings of the 3rd International Workshop on Link discovery LinkKDD*, pages 66–73.

Wang, J., de Vries, A. P., and Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 2006*, pages 501–508, Washington, USA.

Willet, P. (1998). Recent trends in hierarchic document clustering: a critical review. *Information Processing & Management*, 24(5):577–597.

Yarowsky, D. and Florian, R. (1999). Taking the load off the conference chairs: towards a digital paper-routing assistant. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 220–230, Washington, USA.