UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ROBERTO DIAS TORRES JÚNIOR

Combining Collaborative and Content-based Filtering to Recommend Research Papers

Dissertation presented in partial fulfillment of the requirements for the degree of Master in Computer Science.

Prof. Dr. Mara Abel
Advisor

Prof. Dr. John Riedl
Co-Advisor

Porto Alegre, January 2004.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Torres Júnior, Roberto Dias

Combining Collaborative and Content-based Filtering to Recommend Research Papers, Roberto Dias Torres Júnior. - Porto Alegre: Programa de Pós-Graduação em Computação, 2004.

66 p.: il.

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Computação, Porto Alegre, 2004. Orientadora: Mara Abel. Co-orientador: John Riedl.

1. Recommender System. 2. Collaborative Filtering. 3.Content-based Filtering. 4. Personalization. I. Abel, Mara. II. Riedl, John. III. Title.

ACKNOWLEDGEMENTS

I would like to thanks for the people that in one way or another helped me in the conclusion of this important journey of my life. Especially:

Thanks to Eliseo that from the very beginning motivated me, particularly at my trip to the University of Minnesota. Thanks for the entire BDI group: Luis Álvaro, Laura, Kelen, Taísa, Mariângela, Sandro, Eduardo and Felipe. It is nice to have people like you nearby. Thanks for the people at UFRGS, mainly Filipe (Portuga), Fabrício and Fachinello. I'm also glad to Adolfo, who helped a lot while I was at Minneapolis. Thanks too to five people: Davi, Paula, Suzana, Jennifer and Zé. I'm sure you know why.

Special thanks to the people that were with me at the United States. You've made this time much more pleasant, despite of the distance from the family. Thanks for all of GroupLens, particularly Tony, Mamum, Dan Cosley, Dan F and Doreen. Thanks to Eugene and Patricia Allred for hosting me in their home and for the family support. Thanks to Katy, for your friendship since the beginning. Thanks to my friend Sean McNee for all thoughts shared and for being an excellent co-worker. Finally, thanks to Prof. Joseph Konstan for all ideas and participation on my work.

Very especial thanks to my advisors, Mara Abel and John Riedl. During this journey, you've taught me much more than develop a research of quality. I admire you and consider you examples to be followed.

Very especial thanks to my girlfriend, Priscilla, who has making every day of my life better. From the beginning, we knew it was meant to be.

Most of all, thanks to my parents, Roberto and Deolinda, who from my first steps guided me. Due to you I've learn what honesty and dedication is, besides other values that I carry with me. Thanks to my sisters, Débora and Cristina, who always gave me support at the hard times of our lives. Above all, thanks to God for my health and for having the opportunity to meet such wonderful people in my life.

AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas que de alguma forma me ajudaram na conclusão dessa importante jornada de minha vida. Em especial:

Obrigado ao Eliseo que desde o começo me incentivou na pesquisa, principalmente minha ida à Universidade de Minnesota. Obrigado a todos do grupo BDI: Luis Álvaro, Laura, Kelen, Taísa, Mariângela, Sandro, Eduardo e Felipe. É bom ter pessoas como vocês ao redor. Obrigado aos outros colegas da UFRGS, principalmente Filipe (Portuga), Fabrício e Fachinello. Também sou grato ao Adolfo que muito me ajudou enquanto estive em Minneapolis. Obrigado também a cinco pessoas: Davi, Paula, Suzana, Jennifer e Zé. Tenho certeza de que vocês sabem o motivo.

Um obrigado especial às pessoas que estiveram comigo nos Estados Unidos. Vocês com certeza fizeram esse período muito agradável, mesmo estando longe da família. Obrigado a todos do GroupLens, particularmente Tony, Mamum, Dan Cosley, Dan F e Doreen. Obrigado a Engene e Patricia Allred por me receberem em sua casa e pelo suporte familiar. Obrigado à Katy por sua amizade desde o começo. Obrigado ao meu colega Sean McNee por todos os pensamentos compartilhados e por ter sido um excelente colega de trabalho. Por fim, obrigado ao professor Joseph Konstan por todas as idéias e participação no meu trabalho.

Um obrigado especial também aos meus orientadores, Mara Abel e John Riedl. Durante essa jornada, vocês me ensinaram muito mais que desenvolver uma pesquisa de qualidade. Admiro vocês e os considero exemplos a serem seguidos.

Um obrigado muito especial à minha namorada, Priscilla, que tem feito minha vida cada dia melhor. Desde o começo sabíamos que daria certo.

Fundamentalmente, obrigado aos meus pais, Roberto e Deolinda, que desde os meus primeiros passos me guiam. Graças a vocês aprendi o que é dedicação e honestidade, além dos outros valores que carrego comigo. Obrigado também às minhas irmãs, Débora e Cristina, que sempre me apoiaram nos momentos difíceis de nossas vidas. Acima de tudo, obrigado a Deus pela saúde e por me dar a oportunidade de conhecer em minha vida essas pessoas.

CONTENTS

LIST OF ABBREVIATIONS

ACF	Automatic Collaborative Filtering
CBF	Content-Based Filtering
CF	Collaborative Filtering
LSI	Latent Semantic Index
RS	Recommender Systems
TF-IDF	Term-frequency Inverse-document-frequency
WWW	World Wide Web

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The number of research papers available today is growing at a staggering rate, generating a huge amount of information that people cannot keep up with. According to a tendency indicated by the United States' National Science Foundation, more than 10 million new papers will be published in the next 20 years. Because most of these papers will be available on the Web, this research focus on exploring issues on recommending research papers to users, in order to directly lead users to papers of their interest.

Recommender systems are used to recommend items to users among a huge stream of available items, according to users' interests. This research focuses on the two most prevalent techniques to date, namely Content-Based Filtering and Collaborative Filtering. The first explores the text of the paper itself, recommending items similar in content to the ones the user has rated in the past. The second explores the citation web existing among papers. As these two techniques have complementary advantages, we explored hybrid approaches to recommending research papers.

We created standalone and hybrid versions of algorithms and evaluated them through both offline experiments on a database of 102,295 papers, and an online experiment with 110 users. Our results show that the two techniques can be successfully combined to recommend papers. The coverage is also increased at the level of 100% in the hybrid algorithms. In addition, we found that different algorithms are more suitable for recommending different kinds of papers. Finally, we verified that users' research experience influences the way users perceive recommendations.

In parallel, we found that there are no significant differences in recommending papers for users from different countries. However, our results showed that users' interacting with a research paper Recommender Systems are much happier when the interface is presented in the user's native language, regardless the language that the papers are written. Therefore, an interface should be tailored to the user's mother language.

Keywords: Collaborative Filtering, Content-Based Filtering, Hybrid Recommender System, Research Papers.

1 INTRODUCTION

There are many choices that we have to make on a day-by-day basis. We have to choose what food to eat, what movies to watch, what books to buy. After the invention of the Internet, it has just become worse. Much of the information that before was hard to make available everywhere now becomes available almost instantly. Magazines, newsletters, and Usenet news create an information overload that we can't cope with.

One of the first attempts to reduce the information overload was the development of information retrieval systems. These systems respond to a user query, usually based on keywords. Search engines like Google (GOOGLE 2003) and library systems (LANL 2003; NZDL 2003) are examples of these systems.

According to the US National Science Foundation, in 1999 more than 530,000 research papers were published in more than 1,900 journals worldwide. Since 1986 the number of papers published each year has increased at a rate of 1% per year (FOUNDATION 2003). If this trend continues, more than 10 million papers will be published in the next 20 years. Even within fairly narrow fields such as artificial intelligence and human-computer interaction, it is impossible to cope with all work being published — especially interesting interdisciplinary work that may be published in a variety of venues that are not familiar to a researcher in a particular discipline.

In the past decade, Recommender Systems were built. Using a huge amount of available items and knowledge about users' preferences, Recommender Systems have been applied in many domains like Usenet Netnews (Resnick et al. 1994), movies (Hill et al. 1995; Herlocker et al. 1999), audio CDs (Shardanand;Maes 1995), television guides (Cotter;Smyth 2000), and research papers (McNee et al. 2002). In many of these domains, researchers have applied a large number of techniques, each of them with its strengths and weaknesses.

To deal with the large number of available scientific papers, many digital libraries have been built, such as the ISI Web of Knowledge (Knowledge) and LANL (LANL) . Systems like CiteSeer (Bollacker et al. 1999) and the ACM Digital Library (ACM) store research papers in a centralized repository, making them available through parametric searches.

The overwhelming number of research papers available online and the huge amount of papers generated by year make the research papers domain a good target for Recommender Systems. In addition, research papers have two interesting properties: the *text* of the papers themselves, and the *citation web*, which links papers to other relevant papers. These properties can be fully explored by Recommender

Systems´ techniques, either analyzing similar texts based on text or finding relevant papers based on their linking to other papers. Hence, the goal of this dissertation is to analyze several issues regarding the recommendation of research papers. In order to do so, we've defined three hypotheses, described below. We are going to judge our hypotheses as true based on the results of our offline and online experiments.

H1: Collaborative Filtering and Content-Based Filtering can be combined to produce recommendations of research papers.

Collaborative and Content-based Filtering can be individually used to recommend research papers. Collaborative Filtering has already been successfully explored (McNee;Albert et al. 2002) through the analysis of the citation web. The text of the research papers can also be explored to recommend similar papers based on text similarity. In addition, the techniques have complementary characteristics that allow one's advantages overcome the other's disadvantages, and vice-versa. Therefore, we hypothesize that their combination in a hybrid approach will also be successful.

H2: Different algorithms are more suitable for recommending different kinds of papers.

There are different kinds of papers in the literature, such as surveys and novel papers. We also consider in this dissertation that a paper can be: authoritative, specialized, and introductory. As users are interested in different kinds of papers in different situations, would be valuable to find out if different properties of the algorithms might make one algorithm more suitable than another to recommend different kinds of papers.

H3: Users with different levels of experience perceive research papers recommendations differently.

Because the research papers domain has a variety of users, ranging from undergraduate students to professors and researchers, we believe that the level of experience might influence the way users perceive recommendations.

We are also interesting in finding issues on how people from different countries perceive recommendations. Furthermore, we want to investigate language differences. In order to evaluate these issues, we conducted an online experiment with users from different countries and in different language versions.

Because Recommender Systems gained more attention in the past decade and most of the research published on this field has been held in conferences in the United States and Europe, this dissertation has as its goals be as complete as possible. This research was conducted by the Universidade Federal do Rio Grande do Sul, in a strong association with the University of Minnesota – United States. The remaining of this dissertation is as follows: section 2 presents the related work of this dissertation. Section 3 shows the design of the algorithms proposed. Section 4 describes the offline and online experiments along with a discussion about their results. Section 5 describes an analysis and a discussion about all results. Finally, section 6 presents a conclusion and future work.

2 RECOMMENDER SYSTEMS AND RELATED WORK

A Recommender System (RS) is a system that recommends items to users among a huge stream of available items, according to users' interests. An item is anything that a user might have interest in, like a movie, a restaurant or a book. The user's interest in an item can be differently represented: by the use of ratings, which are numeric evaluations that users give to items, or in the case of textual domains, by the similarity between an item and the items the user use to consume.

To store the users' preferences about the items that are of their interests, systems use user profiles. In most Recommender Systems, a user profile is represented by a set of ratings and/or a set of keywords of interest. The ratings are given by users ranging from 1 to 5 or 1 to 7, where the higher the number, the higher is the interest, and keywords are automatically extracted from the texts users read in the past. Ratings are then aggregated through a series of computations to measure users' similarity and then recommend items of interest to them. Texts are matched against the user profile and the texts most similar to the profile are recommended. Also, keywords in the user profile can have weights, indicating how much the user values each keyword.

Ratings can be explicit or implicit. Explicit ratings are generally a single numeric summary rating for each item (Resnick;Iacovou et al. 1994; Shardanand;Maes 1995). Implicit ratings have the advantage of reducing the user's burden to enter ratings, and are generally extracted from purchases records or browsing behavior. Other sources of implicit ratings being explored are the time spent reading (Resnick;Iacovou et al. 1994; Claypool et al. 2001) and URL references in Usenet postings (Terveen et al. 1997). Other browsing behavior indicators like mouse, keyboard and scrollbar activities have also been investigated as implicit interest indicators by Claypool (Claypool;Le et al. 2001).

Regarding its general architecture, a recommender system usually has: (i) background data, which is the information the system has before start the recommendation process; (ii) input data, the information the user has to enter in order to get recommendations; (iii) an algorithm, that combine the input data and the background data to produce recommendations (Burke 2002). In a real system, background data is the user profiles, and the input data are the actions the user performs to get a recommendation. This process is shown in figure 2.1.

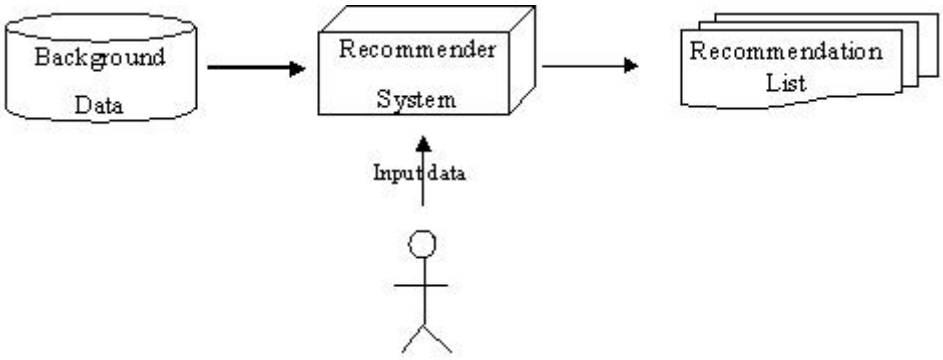

Figure 2.1: Recommender System Basic Architecture

Recommender systems have been applied to many domains and different techniques have been proposed. The systems GroupLens (Resnick;Iacovou et al. 1994), and Ringo (Shardanand;Maes 1995) apply Collaborative Filtering in the domains of, respectively, Usenet News filtering and music. The system Krakatoa (Kamba et al. 1995) applies Content-based Filtering in the domains of online newspapers. Fab (Balabanovic;Shoham 1997), and P-Tango (Claypool et al.), apply a hybrid approach of collaborative and Content-based Filtering to recommend, respectively, web pages, and online newspaper.

The work presented in this dissertation is an extension of the TechLens project (McNee;Albert et al. 2002), that has used Collaborative Filtering on the domain of research papers. Here we extend this research by combining content-based and Collaborative Filtering to recommend research papers. Along this dissertation, the terms “research papers” and “papers” are used interchangeably. Below, we present a taxonomy for hybrid Recommender Systems. We also present collaborative and Content-based Filtering along with examples of systems. Finally, we present some hybrid Recommender Systems.

2.1 Taxonomies of Recommender Systems

There are many taxonomies proposed in the literature: Schafer built a comprehensive taxonomy, taking into account the data, the algorithms and the applications of a recommender system (Schafer et al. 1999). Reategui’s taxonomy regards the way the recommendations should be delivered to users (Reategui;Campbell 2001). Although these taxonomies give a good overview of Recommender Systems’ features and help researchers understand and evaluate RS, they are not appropriate to the hybrid algorithms developed on this research.

Therefore, the algorithms developed in this research are classified based on the taxonomy of hybrid Recommender Systems proposed by Burke (Burke 2002). This taxonomy divides hybrid approaches into seven categories: weighted, switching, cascade, feature combination, feature augmentation, mixed and meta-level. From those, we decided to first explore the two most straightforward classes: *feature augmentation* and *mixed*. In feature augmentation, one technique is employed to a classification of an item and that information is then incorporated into the process of the next recommendation technique. The mixed approach for a hybrid recommender

system is the one that the list of recommendations comes from more than on technique. We believe these classes are a good choice for start exploring hybrid algorithms in this domain.

The other approaches are described as follow. In the *weighted* model, the score of a recommended item is computed from the results of all of the available recommendation techniques present in the system. In the *switching* model, the system uses some criterion to switch between the recommendation techniques. In the *cascade* model, one recommendation techniques is employed first to produce a coarse ranking of candidates and a second technique refines the recommendation from among the candidate set. In the *feature combination*, features from different recommendation data sources are thrown together into a single recommendation algorithm. Finally, in the *meta-level* model, the model learned by on recommender is used as input to another. Table 2.1 summarizes the hybrid models. Models used in this research are bolded.

Table 2.1: Hybrid Models

Weighted	The scores of several recommendation techniques are combined together to produce a single recommendation
Switching	The system uses some criterion to switch between the techniques
Cascade	One recommendation refines the recommendations given by another
Feature Combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm
Feature Augmentation	**Output from one techniques is used as input to another**
Mixed	**Recommendations from several different recommenders are presented at the same time**
Meta-level	The model learned by one recommender is used as input to another

2.2 Collaborative Filtering

The term Collaborative Filtering (CF) was coined by Goldberg in the recommender system Tapestry (Goldberg et al. 1992). Collaborative filtering recommends an item to a user if similar users have liked that item. The intuition behind this is that if users agreed in their likes in the past, they tend to agree again in the future (Resnick;Iacovou et al. 1994).

Society already uses a basic form of Collaborative Filtering: “word of mouth” (Shardanand;Maes 1995; Riedl;Konstan 2002; Miller 2003). For instance, when looking for a restaurant to eat, we rely on friends’ advice. Also, when looking for a book to read, we ask friends who have the same taste we do. The computer’s

role in the process of CF is to make this process automatic. The term Automatic Collaborative Filtering (ACF) has been used to describe systems that automate word of mouth, but in this dissertation we call them only Collaborative Filtering systems.

The most prevalent algorithms used in CF are the neighborhood-based methods (Resnick;Iacovou et al. 1994; Herlocker;Konstan et al. 1999). In neighborhood-based methods, a subset of appropriate users (neighbors) is chosen based on their similarity to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user. There are many approaches for neighborhood-based CF: user-user (Herlocker;Konstan et al. 1999; Sarwar et al. 2000), item-item (Miller 2003), co-occurrences (McNee;Albert et al. 2002). This dissertation uses the user-user approach.

Other algorithmic methods that have been used are Bayesian networks (Breese et al. 1998), singular value decomposition with neural net classification (Billsus;Pazzani 1998), induction rule learning (Basu et al. 1998), and recommendation frames (Reategui et al. 2001). Maltz also defined an Active Collaborative Filtering, where users actively participate in the process of recommending items (Maltz;Ehrlich 1995).

2.2.1 User-User Collaborative Filtering

The User-user Collaborative Filtering algorithm is the standard k-nearest-neighbor. In this algorithm, the problem space can be modeled as a matrix where the rows represent the "users", the columns represent the "items" and the cells represent the ratings that the users gave to items. If a cell is empty, it means that the user didn't rate that item. An example of a rating matrix is shown in table 2.2[1].

Table 2.2: Example of a Rating Matrix

	A	B	C	D	E	F
Mari	1	5		2	4	
Luis	4	2		5	1	2
Edu	2	4	3			5
Lau	2	4		5	1	

The problem is defined as predicting the values of missing cells for the active user. The active user is the user for whom recommendations are being generated. Items that have the highest predicted scores are recommended to the active user.

[1] Adapted from RESNICK, P., et al. GroupLens: An open architecture for collaborative filtering of netnews. Computer Supported Collaborative Work Conference, 1994, Chapel Hill, North Carolina - USA. **Procedings.** 1994.

To predict the missing values, there are roughly two main steps to be accomplished: neighborhood formation and prediction generation (Sarwar;Karypis et al. 2000; Miller 2003).

- **Neighborhood formation**

Neighborhood formation consists of finding the most similar users to the active user based on their past agreement on ratings. A similar user to the active user is the one that has the same opinions about items they have rated in the past. There are many ways to formalize this "agreement" of evaluations, such as the similarity metrics of cosine and Pearson correlation (Miller 2003).

The cosine similarity metric considers each user's ratings as a vector (in the *m*-dimensional product-space) and the proximity between users is measured by the cosine of the angle between the two vectors, which is given by the equation 2.1 (Sarwar;Karypis et al. 2000). In the equation, vectors $\vec{u}$ and $\vec{v}$ represent the ratings of users *u* and *v*.

$$\cos(\vec{u},\vec{v}) = \frac{\vec{u} \bullet \vec{v}}{\left\|\vec{u}\right\|_2 * \left\|\vec{v}\right\|_2} \qquad \text{Equation (2.1)}$$

For instance, measuring the similarity through Cosine among the users, we have that Mari's similarity to, respectively, Luis, Edu, and Lau, are 0.61, 0.96, and 0.78. This means that Mari is very correlated with Edu, and reasonably correlated with Lau and Luis.

The Pearson correlation measures how users correlate with each other user based on past agreement in their ratings, according to the equation 2.2 (Resnick;Iacovou et al. 1994; Shardanand;Maes 1995; Herlocker;Konstan et al. 1999; Sarwar;Karypis et al. 2000). In the equation, $w_{a,u}$ is the weight of the active user *a* to a given user *u*, $r_{a,i}$ is the rating of the user *a* to the item *i*, $\bar{r}_a$ is the average of the ratings of the active user *a*. The weight $w_{a,u}$ indicates the computed users' similarity. All the summations and averages in the formula are computed only over those items that both the user *a* and the user *u* have rated.

$$w_{a,u} = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a) * (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i=1}^{m} (r_{u,i} - \bar{r}_u)^2}} \qquad \text{Equation (2.2)}$$

For instance, measuring the similarity through Pearson correlation among the users, we have that Mari's similarity to, respectively, Luis, Edu, and Lau, are -0.8, 1, and 0. This means that Mari is very correlated with Edu, poorly correlated with Luis, and not correlated with Lau.

The Pearson correlation coefficient returns a number in the range [-1; 1], indicating how much one user agrees with the other user, where –1 indicates that they completely disagree and +1 indicates that they completely agree. Cosine returns a number between [0; 1], where 0 indicates that the users are not correlated at all and

1 indicates that the users are very correlated. Herlocker (Herlocker 2000) proposes two ways to create a neighborhood: maximum number of neighbors, or limits based on correlation weight. In the first, the most *n* similar users are put in the same neighborhood and, in the second, all the users with similarity greater than a threshold form the neighborhood. There is a tradeoff between these two options: while in the first it is more likely to have a neighborhood with enough neighbors, it is not guaranteed a high correlation among them. On the other hand, with the second approach, there might not be enough neighbors, but they will be sufficiently correlated. The recommender engine used in our experiments uses a fixed number of neighbors (Sarwar et al. 2001).

Herlocker's research recommends that in a domain where the rating scale is continuous, Pearson is better (Herlocker;Konstan et al. 1999) to measure users' similarity. Breese evaluated different coefficients (Breese;Heckerman et al. 1998) but we didn't find any work saying which coefficient is better for Boolean domains. As our domain is Boolean, we then decided to start our research using the cosine similarity. Also, to perform the user-user algorithm in our experiments we chose the Suggest recommendation engine (Suggest), developed at the University of Minnesota and freely available for research purposes.

- **Prediction generation**

First of all it is important to define the difference of a *recommendation* and a *prediction*. A recommendation is a suggestion of an item (or a list of items) from the domain. To recommend, for instance, the top-10 best comedy movies, a RS has to look over the entire database of movies to find the top-10. A prediction is the generation of a "predicted value" for one particular item. This predicted value would be the evaluation the user would give to an item, if he/she consumes it. In this case, the user comes to the system and, for instance, asks for a prediction for the movie "Star Wars" and the system returns "4.5". One way for a system to generate recommendations is to just have the system generate predictions for all items and then just filter out items that aren't relevant to the user's search and return a sorted list as a result. This is the way recommendations are given in our experiments.

Independent of the coefficient used to create the neighborhoods, the prediction can be generated as a weighted average of all the neighbors' ratings using equation 2.3 (Resnick;Iacovou et al. 1994; Shardanand;Maes 1995; Herlocker;Konstan et al. 1999).

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} (r_{u,i} - \bar{r}_u) * w_{a,u}}{\sum_{u=1}^{n} | w_{a,u} |} \qquad \text{Equation 2.3}$$

The formula measures how every user rate the item and weights it using the similarity with the active user. The prediction is a number in the range of the ratings.

2.2.2 Collaborative Filtering Systems

Many research projects have explored the potential of CF in Recommender Systems, including: GroupLens (Resnick;Iacovou et al. 1994) , and Ringo (Shardanand;Maes 1995).

The **GroupLens** system is a collaborative system to recommend Usenet Net News to users. Users explicitly rated the news in a 1-5 rating scale and the system aggregate their votes and generate neighborhoods using the Pearson correlation coefficient. Recommendations are given as a weighted average among the neighbors' ratings, according to the equation 2.3. GroupLens is considered the first successful system that employs CF and, although it is not running anymore, it is one of the most cited works in the field.

The **Ringo** system, developed at the Massachusetts Institute of Technology, used the same approach of GroupLens but through a 1-7 rating scale. Ringo also proposed a different coefficient, the constrained Pearson, to compute similarity. This coefficient has the same formula of the original Pearson but instead of using the average of the ratings $\overline{u}_a$, it uses 4, which is the midpoint of its seven-point rating scale. Constrained Pearson performed better than its standard approach, but it reduced its coverage. In Ringo, Users explicitly enter their ratings to get recommendations of audio CDs.

2.2.3 Analysis of Collaborative Filtering

Collaborative Filtering algorithms provide three key advantages to information filtering that are not provided by Content-based Filtering (Balabanovic;Shoham 1997; Herlocker;Konstan et al. 1999): (i) independence of content; (ii) the ability to filter items based on quality and taste; and (iii) the ability to provide serendipitous recommendations.

First of all, because the evaluation of the items is up to the humans, there are no constraints for the kind of item being evaluated. Movies, jokes, recipes, or people could be used as domain of a CF system.

Second, CF systems can enhance the process of filtering by analyzing features that go beyond text analysis. This is due to the fact of involving human subjects in the process of evaluating items. Humans can evaluate if a research paper is authoritative or well-written, which is a very hard task for computers.

Finally, CF can recommend items to users that they don't expect to receive, but are good recommendations (serendipity). This is due to the fact that the similarity is measured between people instead of items. For instance, picture two users A and B that have the same tastes about movies, usually comedies. They rate the movies very highly. At some point, user A rates very highly a drama movie. This drama movie might be a good recommendation to the user B and CF explores this.

On the other hand, CF presents some drawbacks (Balabanovic;Shoham 1997; Claypool;Gokhale et al. 1999; Herlocker;Konstan et al. 1999; Cotter;Smyth

2000; Miller 2003): (i) the first-rater problem; (ii) the *start-up* problem; and (iii) the sparsity problem.

The first problem is inherent to the technique. Because the recommendations are items that similar users have rated, an item cannot be recommended until a user rates that item. The second problem refers to the inability of CF algorithms to give recommendations for users with few ratings. It is due to the fact that a user with few ratings cannot be placed in a good neighborhood, because he/she is not similar to anybody. Finally, the sparsity problem occurs because in a real domain, a user is very likely to rate only a small percentage of the existing items, making it difficult to create neighborhoods due to the lack of overlap of tastes. In online retailers such as Amazon.com (AMAZON) there are millions of books that a user could never possibly rate. It is also true in the research papers domain where thousands of new research papers are published every year (FOUNDATION 2003).

2.3 Content-Based Filtering

The Content-based Filtering (CBF) approach has its roots in the information retrieval (IR) field, and employs many of the same techniques (Balabanovic;Shoham 1997). An item is recommended if this item is similar to the ones the user rated highly in the past. For instance, if a user profile contains the words "knowledge", "discovery" and "rules", a new paper about Data Mining is very likely to be recommended to him/her, because the paper and the user profile have words in common. The intuition behind is that if the user liked an item in the past, he/she tends to like other items with similar content in the future.

In general, a user profile is composed of a set of keywords and associated weights. These weights indicate the strength of that word in the recommendation process. The user profile is matched against a corpus of documents and the most similar ones are recommended. Other user profiles have been studied to personalize content distribution to users, such as in institutional sites (Lima;Pimenta 2002).

There are many ways to compute the similarity between texts (Salton;Buckley 1988; Baeza-Yates;Ribeiro-Neto 1999). However, this goes beyond the scope of this dissertation. For the purpose of this research, it is necessary that a technique takes into account not only the frequency of the words in a document, but also how these words discriminate documents. For instance, the word "Internet" may appear in many documents, though not discriminating any document. However, the word "crossing-over" could not appear in many documents but they are extremely discriminating among documents that relate to genetic algorithms. Therefore, this dissertation uses TF-IDF similarity to measure text similarity. This technique has been widely applied in many domains and is considered a successful technique for text similarity (Kroon et al. 1996).

2.3.1 TF-IDF Content Similarity

The TF-IDF similarity stands for term-frequency inverse-document-frequency and combines the frequency of terms in documents with the distribution of the terms in the whole collection of documents. Theoretically, documents with high number of similar words and also with discriminating words should be the most similar to the query (Salton;Buckley 1988).

In a huge collection of documents, it is very likely that larger documents might be retrieved first, because the term-frequency is higher. To handle that, TF-IDF has a third component: a normalization component, making possible both larger and smaller documents being retrieved, regardless of their sizes.

To perform the TF-IDF over texts, we used the freely available Bow Toolkit, developed at the Stanford University. This is a very powerful library that allows document classification, document retrieval and document clustering (McCallum 1996). This library performs stemming based on Porter's algorithms (Porter 1980) and eliminates stopwords based on the SMART stoplist (Salton;Buckley 1988).

2.3.2 Content-Based Filtering Systems

Many research projects have been using only Content-based Filtering to recommend items. Among them, Krakatoa (Kamba;Bharat et al. 1995) and the one developed by Woodruff (Woodruff et al. 2000).

The **Krakatoa** Chronicle is a personalized newspaper that creates a realistic rendering of a newspaper, with the multi-column format. User profiles are created based on a set of keywords and ratings can be entered explicitly or implicitly (words extracted from news articles the user read). Every word in the news read by the user is added to his/her profile and the weights set accordingly. Words extracted from the news that had explicit ratings receive more weight than words extracted from news not rated. Documents are recommended to a users based on three parameters: the score that each article receives based on the user's profile (through TF-IDF similarity), the average score received by each article over the community of users, and the size and composition of each article. Based on these parameters, each user has access to a personalized newspaper, according to their interests.

Woodruff et al. developed a project held at the Xerox Palo Alto Research Center which goal was to recommend reading material on a digital book (Woodruff;Gossweiler et al. 2000). Authors developed 6 hybrid algorithms to combine analysis of text and the citation web existing between the papers. A spreading activations function was used over the texts to recommend next reading papers. Different combinations of weights given to citation data and texts were tested and the results in their corpus showed that spreading activation functions is more useful than citation data. No ratings needed to be provided by users.

CiteSeer was developed by the NEC Research Institute and it is today the largest Computer Science research paper repository on the Web

(Bollacker;Lawrence et al. 1999; Lawrence et al. 1999). It introduced the automatic citation indexing using a lot of heuristics and machine learning techniques to process documents.

2.3.3 Analysis of Content-Based Filtering

Content-based Filtering can be successfully applied to recommend items in textual domains. CBF presents two key advantages: (i) no first-rater problem, and (ii) no sparsity problem.

The first advantage is because CBF recommends an item to a user if the user's profile and the text of the item share words in common. The second advantage is due to the fact that, in textual domains, for most items can be computed a similarity between its text and the user's profile.

However, a content-based system also has several shortcomings (Balabanovic;Shoham 1997): (i) In some domains, like movies and music, it cannot successfully analyze the content; (ii) because it analyzes the text, it cannot consider aspects like authoritativeness of the author, the writing quality and style; and (iii) the over-specialization problem.

First of all, current technology is not able to analyze successfully video and audio streams. Alternatively, reviews of items (such as movies) have been used, but it has the problem of bias of the reviewers and the reviews are not always available in digital format.

Second, most text analysis techniques are based solely on word analysis. Thus, they do not consider author's style of writing, author's authority in the research field, and text clarity. In addition, many of the techniques don't consider the structures of the text, like title, paragraphs and sections.

Finally, the over-specialization problem refers to the fact that techniques analyzes the content of the texts, then recommending items with similar content, without spreading between other subjects. For instance, if a text uses the word "car" and other text uses the work "automobile", a technique might not consider these two texts similar.

2.4 Hybrid Systems

Taking a closer look at the characteristics of each technique, we can see that they are complementary. The weaknesses of CF are the strengths of CBF and vice-versa. CBF does not suffer from the first-rater problem, as long as a new item can match a user profile based on keywords. In addition, CBF also does not suffer from sparsity, since every item can be related to a user profile (computing the similarity between them). On the other hand, CF does not suffer from content-dependency, since it can be applied to every domain in which humans can evaluate items. Also, CF uses quality and taste when recommending items. Finally, the serendipity of CF guarantees that there is no over-specialization problem.

Therefore, the advantages and disadvantages of both techniques suggest that they can be combined to eliminate both weaknesses. Table 2.3 summarizes their advantages and disadvantages.

Table 2.3: CF and CBF Advantages and Disadvantages

	Advantages	**Disadvantages**
Collaborative Filtering	Content-independent Use of quality and taste Serendipity	First-rater Sparsity problem
Content-based Filtering	No first-rater problem No sparsity problem	Content-dependent Non-use of quality and taste Over-specialization

Hence, the goal of a hybrid recommender system is to combine different techniques to mutually eliminate their drawbacks. There are many ways how different filtering systems can be combined. Burke identified 53 different possible hybrids from which only 14 have been explored (Burke 2002). Case-based Reasoning (CBR) has also been combined with CF (McGinty;Smyth 2001).

Theoretically, the only problem that a CF-CBF hybrid approach does not address is the start-up problem. Solutions to this problem have been proposed by Rashid (Rashid et al. 2002) and by demographic Recommender Systems (Pazzani 1999). Below we describe a few hybrid Recommender Systems developed to date.

Fab is a hybrid recommender system developed at Stanford University which recommends web pages to users (Balabanovic;Shoham 1997). It uses content analysis to create the user profiles and compare these profiles to determine similar users for collaborative recommendation. Fab's architecture is basically formed by two kinds of agents: collection agents and selection agents. While the collection agents are responsible for gathering interesting pages on the Web, the selection agents are responsible for redirecting these interesting pages to the appropriate users. The users can also rate every recommendation received on a 7-point scale. These ratings are used to update the user's personal selection agent and also the collection agents. According to Burke's taxonomy, Fab is a meta-level hybrid model, where the model generated by one technique is used as input of another.

P-Tango is a hybrid recommender system developed at the Worcester Polytechnic Institute applied to recommend news in an online newspaper (Claypool;Gokhale et al. 1999). It uses a weighted average of the content-based prediction and the collaborative prediction. The users rate items explicitly. Also, the weight given to each technique is user-dependent: every time a user rates an item, the absolute error of the CF and the CBF techniques are measured and the weights are set accordingly. Following Burke's taxonomy, P-Tango is a weighted model.

Content-Boosted Collaborative Filtering (**CBCF**) is a technique developed at the University of Texas to combine Content-based Filtering into the process of Collaborative Filtering. Following the Meta-level Burke's model, it is

applied in the domain of movie recommendations. The similarity between the user profile and every movie the user has not rated is measured and this similarity is used to fill the cells in the rating matrix. Collaborative Filtering is applied over this denser matrix using Pearson correlation coefficient. The users had to enter their ratings in an explicit way. CBCF follows the meta-level hybrid model (Melville et al. 2001).

Developed at the University of Dublin, Ireland, the **PTV** System is a recommender for TV guides (Cotter;Smyth 2000), both in digital TV guides and Wireless devices. User profile is represented by attributes of the TV shows they have watched, like casting actors and directors. PTV follows Burke's mixed model, generating a final recommendation list with items coming either from CF and CBF. Despite users' profiles are created automatically, users can manually change their profiles through direct feedback. Table 2.4 shows all systems presented in this section along with their characteristics.

Table 2.4: Recommender Systems

	Techniques	**Domain**	**Ratings**	**Model**
GroupLens	CF (Pearson)	Usenet Net News	Explicit	-
Ringo	CF (Constrained Pearson)	Audio CDs	Explicit	-
Krakatoa	CBF (TF-IDF)	Newspaper	Explicit Implicit	-
CiteSeer	Citation Indexing	Research Papers	-	-
Woodruff's	Spreading Activation Function	Digital Book	-	-
Fab	CF, CBF	Web Pages	Explicit	Meta-level
P-Tango	CF, CBF	Newspaper	Explicit	Weighted
CBCF	CF, CBF	Movies	Explicit	Meta-level
PTV	CF, CBF	TV Guides	Explicit Implicit	Mixed

Although many systems have already developed a hybrid approach for Recommender Systems, none of them have explored a hybrid approach of CF and CBF in the domain of research papers. Research papers have characteristics, like the full text of the papers and the citation web existing among them, that make this domain suitable for Recommender Systems. In addition, the huge amount of papers already existing on the Web and the growing number of papers published every year make this domain still more exciting for research.

3 DESIGN OF ALGORITHMS FOR PAPER RECOMMENDATION

This section describes all algorithms implemented in this dissertation, following Burke's classification of hybrid algorithms (Burke 2002). Ten different algorithms were developed; of them, five are hybrids and five are non-hybrids, with the hybrids following Burke's feature augmentation and mixed models. Each algorithm receives one paper as input and generates a list of papers as recommendations. In addition, algorithms running only CF and only CBF were also tested and their results are presented as a baseline comparison.

CBF was applied over the text of the papers while CF was applied over the citation web existing among papers. In order to perform CF in the domain of research papers, a rating matrix had to be created. This was done by mapping the citation web onto a rating matrix (McNee;Albert et al. 2002). In addition, to perform our algorithms, we used two engines, detailed below.

- Building a Rating Matrix

Collaborative Filtering algorithms recommend items based on similarity among users, and this similarity is measured by their profiles. The users' profile in CF systems, as shown in section 2, is represented by a rating matrix. Therefore, it is necessary to create a rating matrix to recommend research papers using CF. We decided to use the same approach used in (McNee;Albert et al. 2002). It considers the papers as "users" and their citations as "items" they have rated. This approach does not suffer from the startup problem, because the citations of the papers populate the rating matrix. It is also expected that the citation lists will be of high quality, which frees the system from ratings consistency. On the other hand, a user cannot add ratings as in other CF domains, because a paper has a fixed set of citations.

Considering the user profile, there are basically three approaches to fill the rating matrix, regarding the gathering of information and the user interests: implicitly for long-term interests, explicitly for long-term interests and explicitly for short-term interests. A user profile can be built containing all papers a user has read in the past. This approach gathers implicit information for long-term user interests. Although keeping track of the user's reading habits over time, this approach does not guarantee that the user has effectively read the paper. In addition, a system has to be built to gather the user's preferences of reading.

A second alternative considers the user profile regarding the users' areas of interest (all papers by field). This approach gathers explicit information from

the user and also requires a system that keeps track of all the papers and asks the user to which field he wants to get recommendations for. The system would only use papers belonging to a selected field when generating recommendations. Despite being interesting in a real system, it has many issues regarding how to classify the papers to the fields, going beyond the scope of this research.

The third approach considers only one paper as the user profile. This approach gathers information explicitly and recommends for short-term interests. It has many scenarios of usage, like a user looking for similar papers to one he/she is reading. We decided to use this approach in this research. Also, this approach allows us to evaluate the algorithms' performance without needing to build a system. In addition, the other approaches require recommendations being generated considering more than one paper, which is considered future work in this dissertation. Table 3.1 summarizes the user's profile alternatives.

Table 3.1: User's Profile Alternatives

Approach	Gathering Information / Users' Interests	Information Used	Advantages	Disadvantages
All Papers	Implicit and Long-term interests	All papers read in the past	Keep track of user's reading habits over time	Requires system
All Papers by Field	Explicit and Long-term interests	All papers read in the past	Filter recommendations based on user's field interests	Requires system
One Paper	Explicit and Short-term interests	Only one paper of interest	Does not require a system	Does not keep track of reading habits over time

- Algorithms' Engines

All of the algorithms implemented in this dissertation consider that there is an engine responsible for giving recommendations for CF and CBF algorithms. This engine tasks are to get the input data of the user and combine it with the background data to produce recommendations, according to figure 2.1 (section 2). Each engine has an input and an output format, shown in figure 3.1.

CBF engine receives a query (set of words) to search for similar documents and returns a list of similar documents along with their similarity to the query. CF engine receives the paper along with its citations and returns a list of recommendations. In our CBF algorithms the query is always the title and abstract of the paper. This was done to eliminate the user's burden to enter keywords of interest and to take advantage of the whole information available in the paper abstract.

Figure 3.1: Input and Output of CF and CBF Engines

3.1 Baseline Algorithms

These algorithms are not hybrid and they run pure CBF or pure CF, both with a few variations. The results were used as a baseline comparison for the hybrid algorithms.

3.1.1 Only Content-Based Filtering Algorithms

This kind of algorithm consists of recommending papers to a user based on content similarity. For a given paper, the most similar papers are recommended. The technique used to compute text similarity was TF-IDF, previously described in section 2. Three different algorithms were built, named as Pure-CBF, CBF-Separated and CBF-Combined:

- Pure-CBF

This algorithm is the most straightforward and for a given paper, it searches for similar papers based on their text similarity. The most similar papers are recommended.

- CBF-Separated

This algorithm is an extension of the Pure-CBF and it explores not only the text of the paper, but also the text of the papers it cites. The algorithm searches for similar documents of the paper itself and for every citation of this paper. For instance, for the paper P the algorithm generates a list of similar papers L_P, and for every citation (C_1, C_2, …, C_n) of the paper P, it generates a list of similar papers (L_{C1}, L_{C2}, …, L_{Cn}). All lists are merged into one single list, sorted based on the returned similarity coefficient. Papers are discarded if they have already been added to the resulting list. Papers with the highest similarity scores are recommended. This process is shown in figure 3.2.

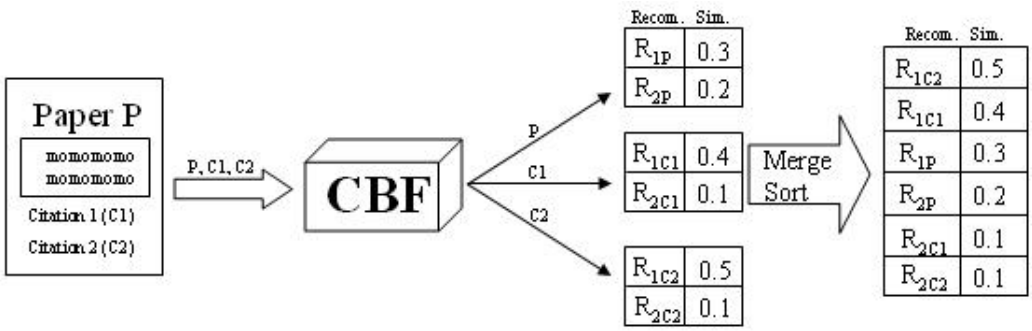

Figure 3.2: CBF-Separated Algorithm

Recommending papers similar to the citations of a given paper should reduce over-specialization, since this algorithm spreads the search space to the contents of the citations too.

- CBF-Combined

CBF-Combined is an extension of CBF-Separated. Instead of generating one list of similar papers for every citation, this algorithm merges the text of the paper and the text of all of the papers it cites together into one large chunk of text. This larger text is submitted as the query to search for similar papers. The most similar papers are then recommended.

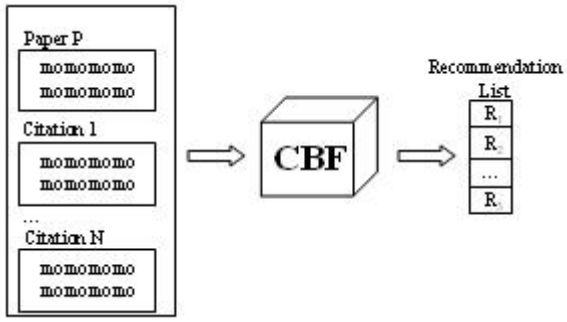

Figure 3.3: CBF-Combined Algorithm

The presence of more words in a single query to the CBF engine should more effectively return similar papers based on content. Figure 3.3 shows the process.

3.1.2 Only Collaborative Filtering Algorithms

This class of algorithms uses only Collaborative Filtering to recommend papers by exploring the citation web existing among papers. Two different algorithms using User-user CF were developed: Pure-CF and Denser-CF.

- Pure-CF

Pure-CF is the standard k-nearest-neighbor User-user CF algorithm (Herlocker;Konstan et al. 1999). It takes the citations of the active paper as input and gives a list of recommended papers as output.

- Denser-CF

Collaborative Filtering has the problem of sparsity (table 2.2). It happens because among the items (papers) available, the user (paper) cites only a few of them, generating a very sparse matrix. Because every citation is actually a paper, it also cites other papers. The idea behind Denser-CF is to use the citations of every citation in the process of recommending papers. For instance, if the paper W cites (A, G), A cites (B, C), and G cites (P, Q), the citations of A and G will augment the citation set of W. Therefore, W would have its original citations (A, G), plus the citations of its citations (B, C, P, Q). According to Figure 3.4, the paper W would have 6 citations instead of 2. This reduces the sparsity of the matrix and might increase the quality of the recommendations.

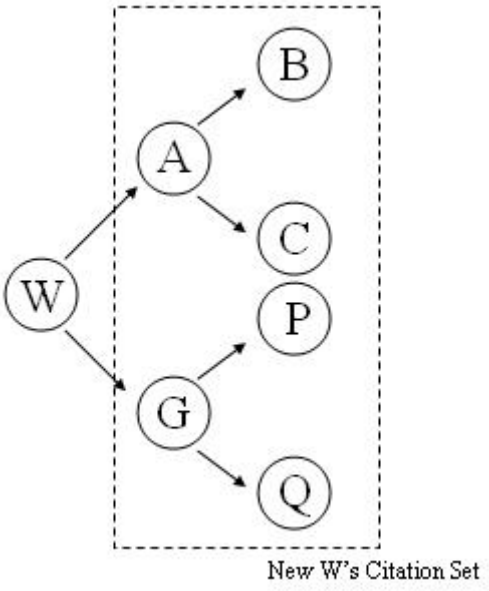

Figure 3.4: Denser-CF Method

3.2 Hybrid Algorithms

As previously explained in this dissertation, CF and CBF together can mutually overcome their shortcomings. Collaborative Filtering can eliminate the over-specialization, content-dependency and the non-use of aspects like quality and taste problems of CBF while Content-based Filtering can eliminate the first-rater problem, and the sparsity of CF.

Four feature augmentation algorithms were developed, that alternatively run either CF or CBF first, using the output of one as input of another. Our algorithms have modules that performed CF or CBF in the input data. Up to 20 recommendations are sent from one to the next module. These feature augmentation algorithms are attractive because theoretically they offer a way to improve the performance of a core system without modifying each module individually (Burke 2002). Table 3.2 shows nine possibilities of feature augmentation algorithms, combining CF and CBF. From those, four were implemented, using different combinations of the single algorithms Pure-CF, CBF-Separated, and CBF-Combined. In the cells there are the algorithms' names and the number of the section in which they are explained.

Table 3.2: Feature Augmentation Hybrid Algorithms

	Pure-CF	**CBF-Separated**	**CBF-Combined**
Pure-CF		CF – CBF Separated (3.2.1)	CF – CBF Combined (3.2.2)
CBF-Separated	CBF Separated – CF (3.2.3)		
CBF-Combined	CBF Combined – CF (3.2.4)		

Another hybrid approach developed is the fusion algorithm, which is a mixed model according to Burke's taxonomy. Fusion runs CBF-Separated and Pure-CF in parallel, differently than the feature augmentation algorithms, which run them in sequence.

3.2.1 CF – CBF Separated Algorithm

In this algorithm, the recommendations from Pure-CF are used as input to CBF-Separated. For every recommendation from CF, the CBF module recommends a set of similar papers (up to 80). Because the recommendations generated by the CF module in order, the recommendations generated by the CBF module have to be scaled by this ordering. Thus, these CBF recommendations are weighted accordingly, with the first set generated from the top CF recommendation receiving weight 1 and the following sets' weights decreased by 0.05 accordingly. The similarity scores of CBF recommendations are multiplied by these weights in descending order. The final similarity of the CBF recommendation, after the weighting process, is computed according to the following formula:

$$Sim(r_j) = Sim(r_j) * (weight(p_i\text{-}1) – weight\_decrement), \text{ where:}$$

- r_j is the j^{th} recommendation in the CBF list.

- p_i is the i^{th} recommendation in the CF list
- $Sim(r_j)$ is the similarity of r_j with p_i
- $Weight(p_{i-1})$ is the weight given to the previous recommendation of CF
- Weight_decrement is the value decremented from the previous weights

This algorithm's recommendation process is shown in figure 3.5. The rectangles with dashed lines show the different modules of this hybrid algorithm.

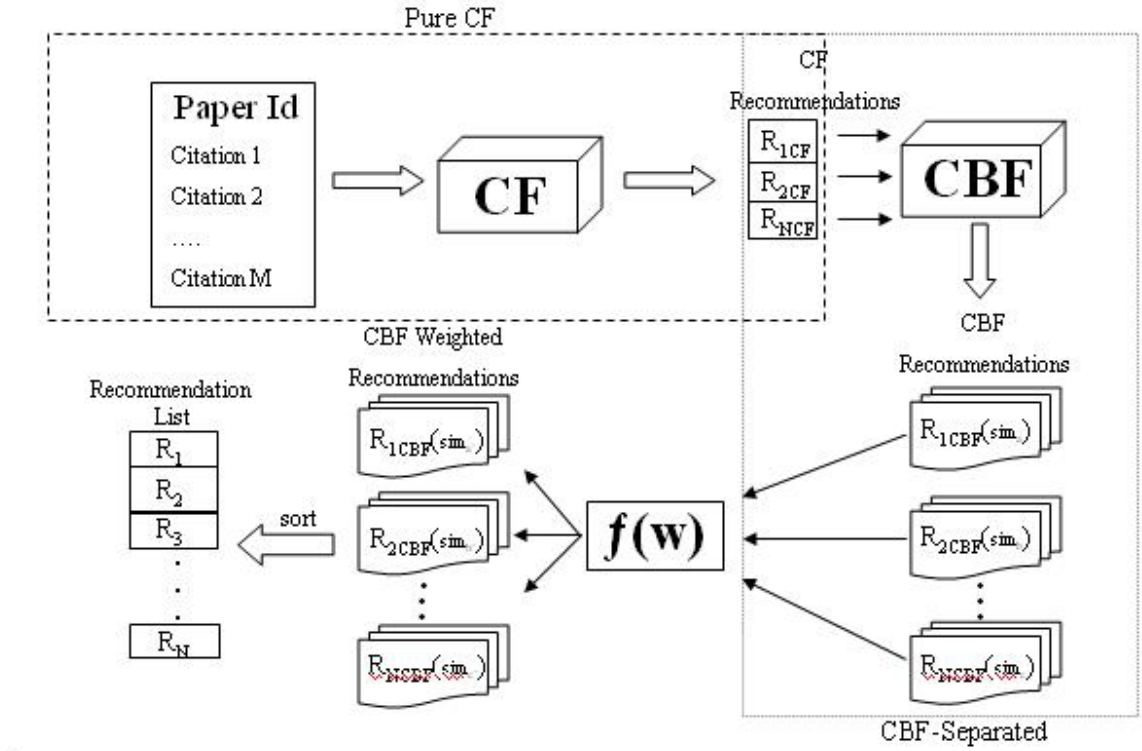

Figure 3.5: CF – CBF Separated Algorithm

For instance, if the initial weight is 1 and the weight_decrement is 0.05, the similarity of similar documents of the R_{1CF} is multiplied by 1, the similarity of similar documents of R_{2CF} will be multiplied by 0.95 and so forth. The weighting function is represented in the figure by *f(w)*.

3.2.2 CF – CBF Combined Algorithm

This algorithm is similar to CF-CBF Separated. However, instead of recommending a set of papers for every recommendation received from the CF module, the CBF module aggregates the text of all of the recommendations given by CF and uses this large chunk of text as its input to CBF (CBF-Combined). The results are sorted by similarity. Figure 3.6 shows the algorithm process.

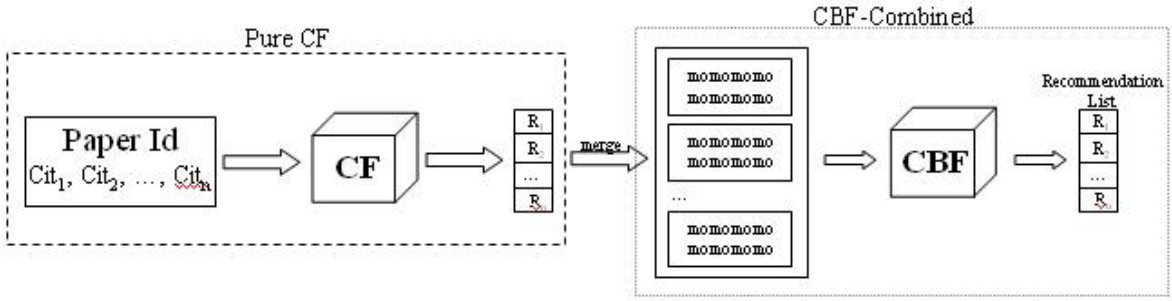

Figure 3.6: CF – CBF Combined Algorithm

3.2.3 CBF Separated – CF Algorithm

Here, CBF-Separated generates recommendations for the active paper. These recommendations are used to augment the active paper's set of citations. The active paper with its augmented set of citations is used as input to Pure-CF to generate recommendations. The recommendation process is shown in figure 3.7.

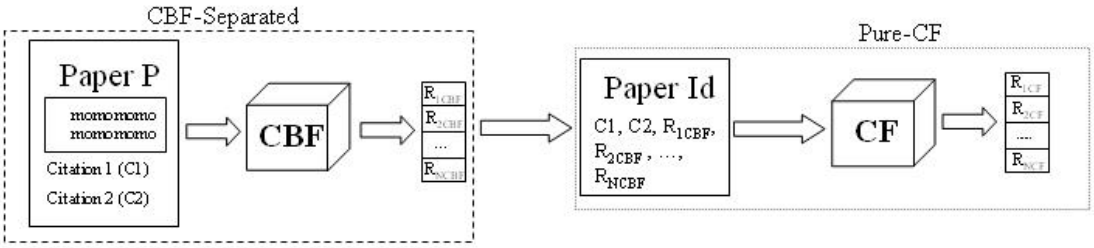

Figure 3.7: CBF Separated – CF Algorithm

3.2.4 CBF Combined – CF Algorithm

This algorithm is identical to CBF Separated – CF, except that CBF-Combined is used in place of CBF-Separated. The process is shown in figure 3.8.

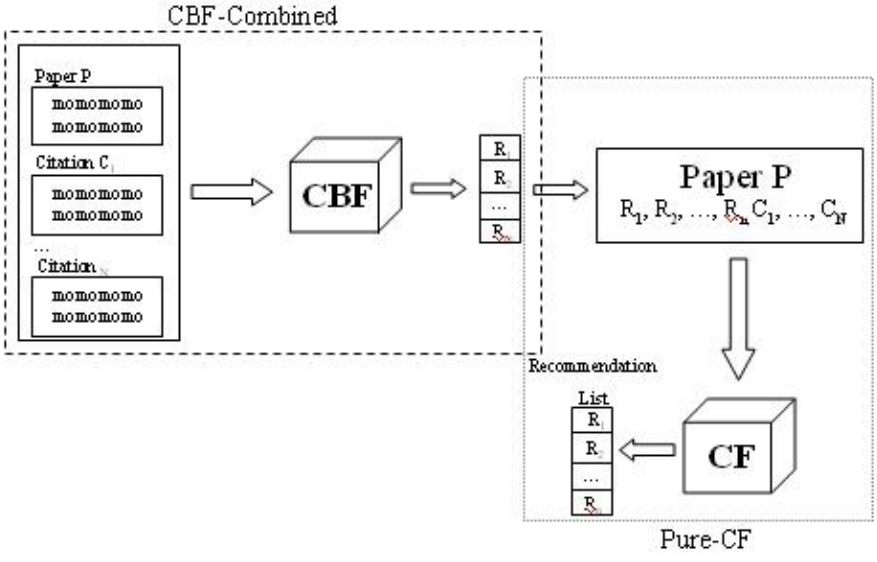

Figure 3.8: CBF Combined – CF Algorithm

3.2.5 Fusion Algorithm

Fusion, our mixed hybrid algorithm, runs the two modules in parallel and generates a final recommendation list by merging the results from both modules. The generation of the final recommendation list is as follows: every recommendation that is present in both modules' result lists is added to the final list with a rank score. This score is the summation of the ranks of the recommendation in their original lists. The final recommendation list is in ascending order based on these scores. Therefore, a paper that was ranked 3rd from the CF module and 2nd from the CBF module would receive a score of 5. The lower the score, the closer to the top an item goes in the final recommendation list. The other recommendations that don't appear in both lists are alternatively added in the final list, coming either from the CF or the CBF recommendation list. The general process of the fusion algorithm is shown in figure 3.9.

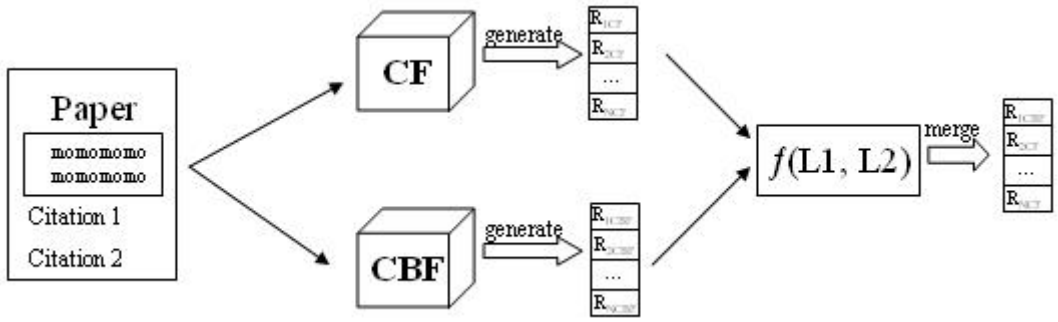

Figure 3.9: Fusion Algorithm

For instance, if CBF recommends the list of papers L_1 = (A, D, H, K) and CF recommends the list of papers L_2 = (D, P, K, J), fusion algorithm will generate a list L_T=(D, K, P, A, J, H). The element D is the first in the final list because its score (3 = 2 + 1) is the lowest of items present in both lists. The element K is the second because its score (7 = 4 + 3) is the second lowest. After that, the resulting list is generating picking one element from each algorithm, starting by CF, and adding to the resulting list. A similar algorithm has been developed by Cotter (Cotter;Smyth 2000).

4 EXPERIMENTS FOR ALGORITHMS' VALIDATION

To validate the hypotheses of this dissertation two kinds of experiments were performed: one **offline** (without users) and one **online** (with users). These experiments have different goals: the offline tries to measure the ability of the algorithms to recommend papers. The online experiments assess users' satisfaction and perceptions about the recommendations they received. We are going to judge our hypotheses based on the results of these experiments.

4.1 DATASET DEFINITION

When describing the dataset, we draw a subtle but important difference between a paper and a citation. A citation is a paper for which the text is not available. A citation therefore is a pointer to a paper. On the other hand, a paper is a citation for which we also have its text. This is important because many citations are references to papers that we do not have in digital format. The text of a paper is represented by its title and abstract. Although many papers are not digitally available, in the future it is expected that the percentage of papers available online will increase, since authors and digital libraries are making them quickly available.

In order to test our algorithms we created a dataset with papers extracted from CiteSeer (Bollacker;Lawrence et al. 1999), an online repository of computer science research papers. This dataset initially had over 500,000 papers and 2 million citations. We limited this dataset in two ways. First, we removed papers that cited fewer than 3 other papers, as we believe these loosely connected papers introduced noise to the dataset. Second, we removed the citations for which we did not have the full paper in our dataset. Therefore, every citation is also a paper in our dataset. We performed this trimming so that both CF and CBF would be able to analyze every item in our dataset. The pruned dataset has 102,295 papers with an average of 14 connections per paper. We define the number of connections of a paper in our dataset as the number of citations it makes plus the number of papers that cites it. To test our algorithms quickly, we also had a smaller dataset with 1173 papers and an average of 8 connections per paper. These dataset will be referred as *full* and *small* datasets respectively.

4.2 Offline Experiments

Offline experiments test whether the algorithms are useful for predicting relevant papers for a given paper. For every paper in the dataset, one citation was randomly removed and the algorithms were used to try to recommend that removed citation. This "Leave one out" methodology has been used in CF offline experiments before (Breese;Heckerman et al. 1998; McNee;Albert et al. 2002). Figure 4.1 shows the process.

We divided the dataset into training and test datasets at a 90% to 10% ratio. Every paper in the test dataset has one randomly removed citation. Ten different training and testing datasets were created for 10-fold cross validation.

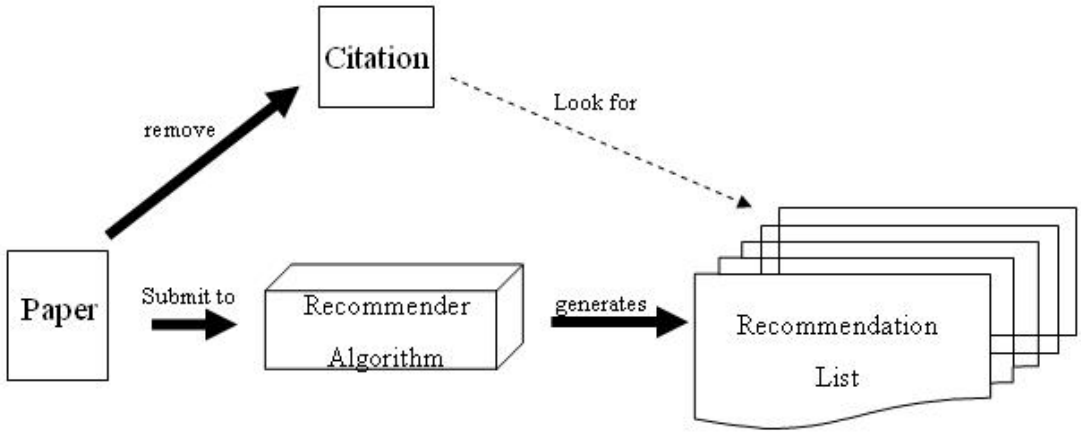

Figure 4.1: Test for Algorithm Performance

This method of experimentation has some limitations. The recommender algorithms could recommend a paper that didn't exist at the time the active paper was published. To handle that, we filtered out recommendations with a publication year later than that of the active paper. It is also important to point out that the algorithms could recommend papers that are very similar to or even better than the removed citation and this might diminish the algorithms' performance, since these citations will appear in the recommendation list before than the removed citations. Even though this is a possibility, we expected the removed citation to be recommended.

4.2.1 Evaluation Criteria

We define "hit-percentage" (HP) as a metric to measure the percentage of the time the recommender algorithm correctly recommends the removed citation. We also measured the rank where the removed citation was found in the recommendation list. Because we believe that rank is important to users, we segmented our analysis into bins based on rank where lower is better. Thus, recommendation in the top-10 bin is better than a recommendation in the top-40 bin.

Recommendations beyond the 40th position are considered "all" because users are not likely to see items recommended beyond this position. Table 4.1 shows the rank layers.

Table 4.1: Rank analysis

Rank of the citation found	Layer
1	top-1
1-10	top-10
1-20	top-20
1-30	top-30
1-40	top-40
1-N	All

From these metrics, we focused on two particular criteria. As we think that users like the best recommendations first, the first criterion is the algorithm's performance in the top-10 HP. The second criterion is the algorithm's ability to recommend the removed citation independently of its rank. It is measured by the "all" HP. A third criterion is used when the top-10 of one algorithm is better than the top-10 of another algorithm but the "all" is not, or vice-versa. In this case, the results of top-1 decide the best algorithm. The best algorithms in the full dataset will be selected to be tested online.

4.2.2 Baseline Algorithms' Results

These algorithms either run only content-based or only Collaborative Filtering. All algorithms described in section 3.1 were tested.

- Only Content-based Filtering

Three algorithms were tested: Pure-CBF, CBF-Separated and CBF-Combined. The results in the full dataset are shown in figure 4.2.

Pure-CBF was worse in all bins, except in the top-1, which it was slightly better than CBF-Combined. CBF-Combined, in the full dataset, performed best in the top-10 (25%) but not in the "all" (50%). CBF-Separated was the best in the "all" (53%) and also in the top-1 (6%). Therefore, based on the third criterion, CBF-Separated was the best in the full dataset.

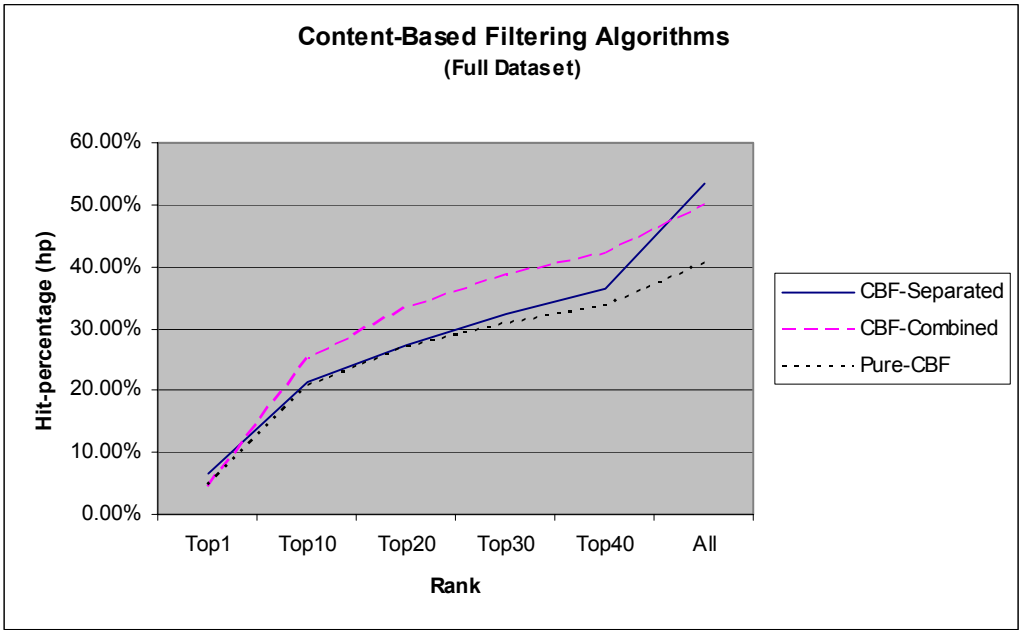

Figure 4.2: CBF Algorithms – Full Dataset

On the other hand, in the small dataset, Pure-CBF had the best top-10 (43.4%), against CBF-Combined (42.5%) and CBF-Separated (39.4%) of the other algorithms. By contrast, it is still the worse in the "all", with 76%, against CBF-Separated (80%) and CBF-Separated (90%). Our hypothesis is that Pure-CBF is not able to give good recommendations in large datasets, where the variation of subjects is much larger than in the small dataset. In addition, all algorithms had a higher average hit-percentage in the small dataset (approximately 80%) than in the full dataset (approximately 50%). This suggests that the high distribution of content in larger datasets limits the ability of CBF recommendation. The results in the small dataset are shown in figure 4.3.

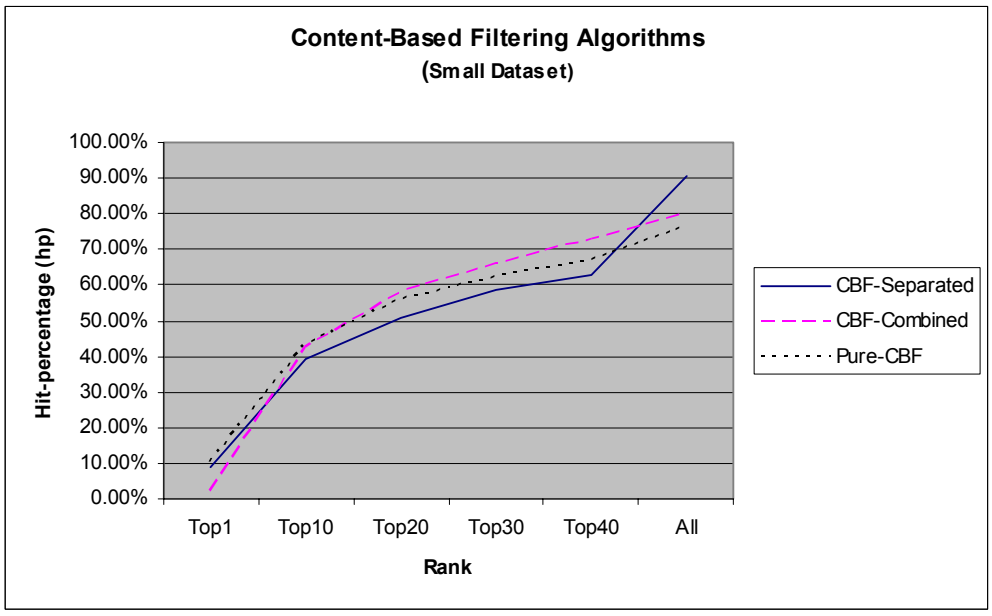

Figure 4.3: CBF Algorithms – Small Dataset

- Only Collaborative Filtering

Two algorithms were tested: Pure-CF and Denser-CF. Results were surprising because it was expected that the denser rating matrix used by Denser-CF algorithm would improve the CF performance. However, in both datasets, Pure-CF performed better. Figure 4.4 shows the results.

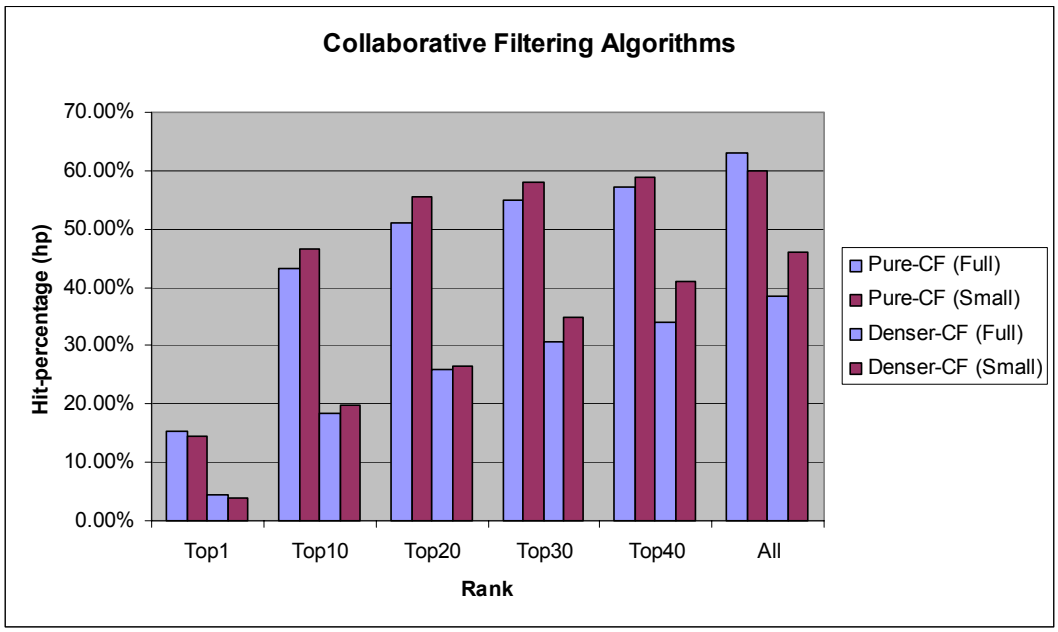

Figure 4.4: CF Algorithms

4.2.3 Hybrid Algorithms' Results

The hybrid algorithms developed in this research followed the feature augmentation and mixed models (Burke 2002). For the feature augmentation algorithms, four different algorithms were tested: CF – CBF-Separated, CF – CBF-Combined, CBF-Combined –CF and CBF-Separated – CF. These hybrid algorithms were running the three standalone versions of Pure-CF, CBF-Separated and CBF-Combined:.

In the full dataset, CBF Combined – CF performed best until top-40. CF-CBF Separated performed better for the "all". Because Pure-CF has its inherent characteristic of finding papers not closely related in content to the active paper, and also CBF-separated spreads in content too, these two algorithms together could together achieve a better performance in the all hit-percentage. CF-CBF Combined performed very poorly. CF-CBF Separated had a high jump from the top-40 to the "all" percentages. The results are shown in figure 4.5.

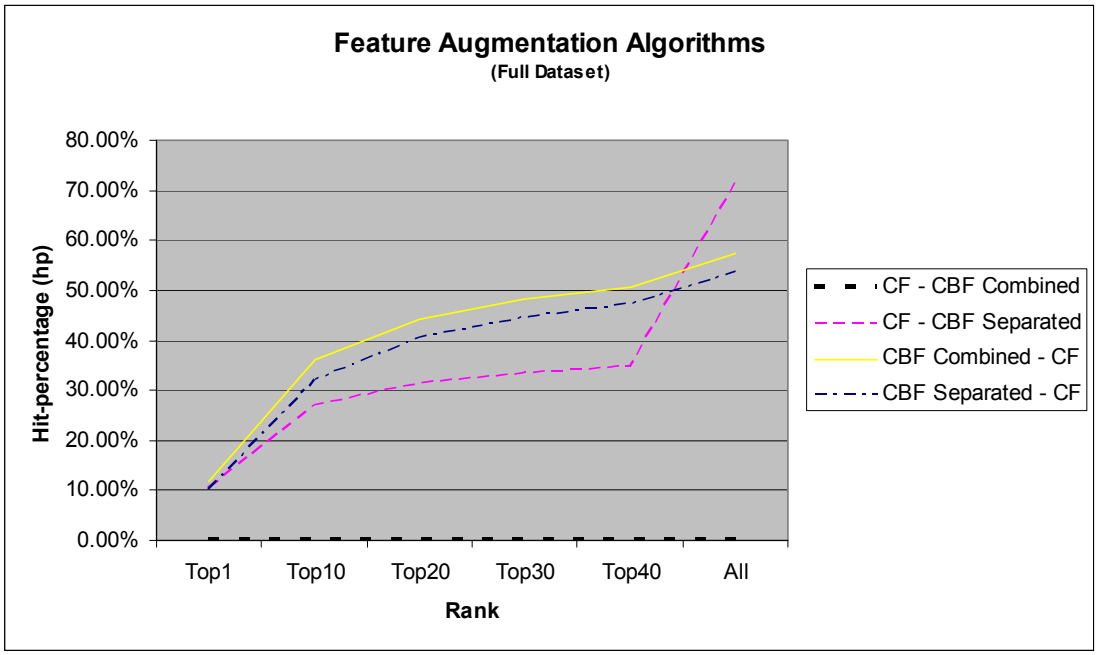

Figure 4.5: Feature Augmentation Algorithms – Full Dataset

In the small dataset, feature augmentation algorithms that started with CF were, respectively, best and worst. CF-CBF Separated performed best, from the top-1 to "all" while CF-CBF Combined performed worst in all bins. CBF Separated – CF is, in all of the times, approximately 4% better than CBF-Combined – CF. Again, CF – CBF Combined was bad. Results are shown in figure 4.6.

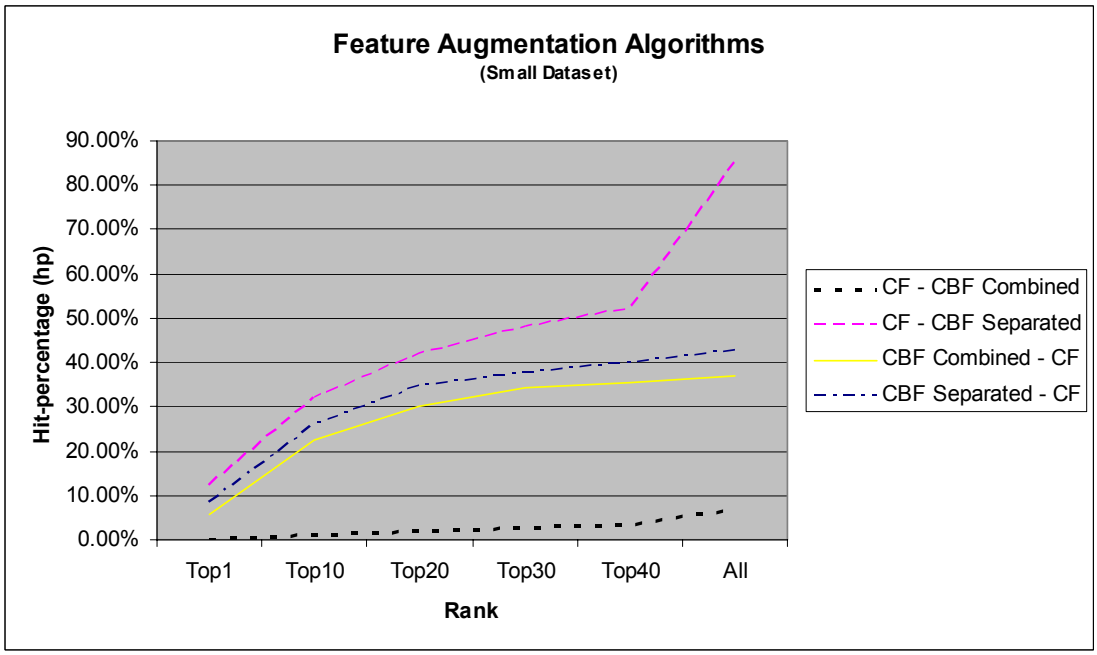

Figure 4.6: Feature Augmentation Algorithms – Small Dataset

The other type of hybrid algorithms is the mixed typed, with Fusion as the only representative. The performance is almost the same in both datasets, except in the "all" analysis. In addition, Fusion presents a very high percentage in the top-1 analysis in both datasets (≈30%). This is a very important characteristic, since users expect to get good recommendations first. Results are shown in figure 4.7.

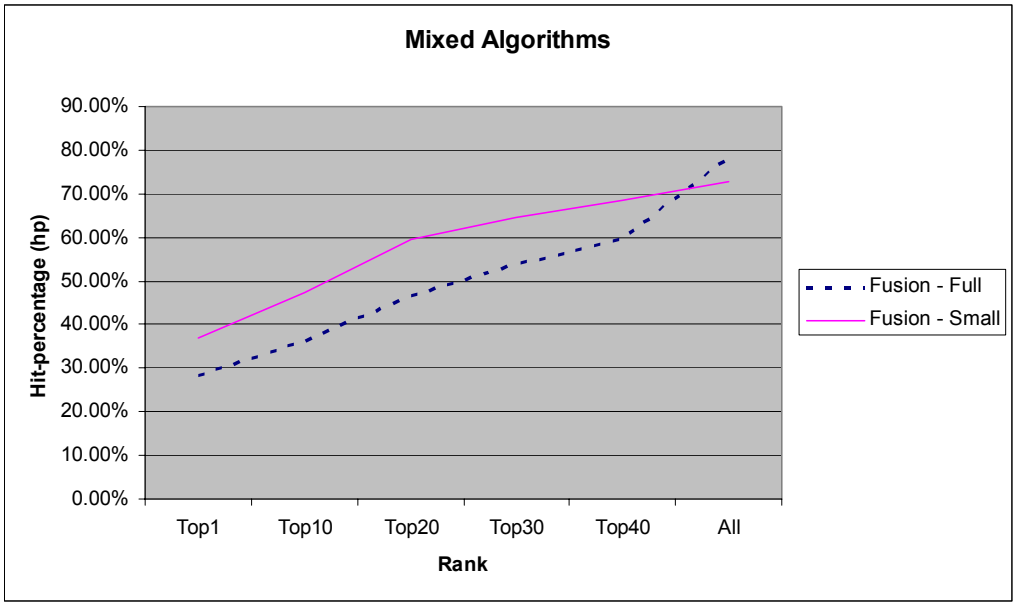

Figure 4.7: Mixed Algorithms

4.2.5 Analysis of the Results

Our analysis considers the algorithm's results obtained in the full dataset. This is the dataset used in the online experiment. Based on this and in the criteria specified in section 4.2.1, the best hybrid algorithms in the full dataset were Fusion, CBF Combined–CF, and CF-CBF Separated. The best non-hybrid algorithms were CBF-Separated and Pure-CF.

Regarding the standalone algorithms, CBF-Separated was expected to perform better than Pure-CBF because it recommends papers similar to the citations of a paper, spreading over the content and reducing over-specialization. However, Pure-CF was not expected to perform better than Denser-CF because the denser rating matrix was expected to overcome the sparsity problem of CF approaches.

Fusion was significantly better compared to every other algorithm at both top-1 (28%) and *all* hit percentage (78%). Hence, Fusion is expected to perform well online. Pure-CF usually had the first or second best performance among all bins.

The other hybrid algorithms performed fairly well. CBF Separated – CF was slightly inferior to the CBF Combined – CF. On the other hand, CF – CBF Combined had a very poor performance compared to every other algorithm. Its all hit-percentage was 0.1%.

Coverage is the percentage of items for which the system could generate a recommendation (Herlocker;Konstan et al. 1999). All of the algorithms had

100% coverage, except Pure-CF which had coverage of 90% in the small dataset and 93% in the full dataset. This high coverage is due to the presence of the CBF approach and it is a significant advantage of building a hybrid algorithm.

4.3 Online Experiment

To assess users' perceptions about the recommendation, we developed an online experimental system, called **TechLens**[+2], consisting of a six-page Web-based experiment where users evaluated recommendations of research papers. Users were asked questions for each individual recommendation they received, and about the set of recommendations as a whole. TechLens[+] had two versions: one in English and the other in Portuguese. TechLens[+] had three main goals:

A) Algorithms' Recommendations Quality Test

Even though the offline experiment is a good estimator of the algorithms' utility, only online experiments will provide real information about the users' perceptions about the quality of the recommendations.

B) Cross-cultural analysis

According to the National Science Foundation, the amount of international cooperation in research is increasing, with more than one-third of all coauthored articles having authors from multiple countries during the years of 1986 and 1999 (FOUNDATION 2003). In this world where international research is becoming more and more important, we wanted to explore any cross-cultural issues in recommending research papers. Thus, when designing our online experiment, we wanted to test for perceived differences in recommendation quality among users from different countries.

Users were invited to participate through links at the Penn State mirror of CiteSeer (PENN 2003) and EBizSearch (EBIZSEARCH 2003). Users were also invited through messages posted in e-mail lists, such as internal lists of the Computer Science departments at: Universidade Federal do Rio Grande do Sul, University of Minnesota, Georgia Institute of Technology, and UC Berkeley. Additional e-mail invitations were sent to the UC Berkeley Collaborative Filtering Interest List[3], the User Modeling Interest List[4], and interest lists of the Brazilian Computer Society[5]. Users ranged from undergraduate students to professors and professional researchers. Because of the nature of the e-mail lists and websites we chose for recruiting users, we expected to have knowledgeable subjects all of whom would be able to complete the experiment. The subjects belonging to the lists, particularly the university internal lists, ranged from many fields of computer science.

---

[2] TechLens is the original project held by the GroupLens Research Group that used only CF to recommend papers. We add the "+" because the "CBF" technique was incorporated in the experiments.

[3] http://www.sims.berkeley.edu/resources/collab/

[4] http://www.um.org/

[5] http://www.sbc.org.br/

The following algorithms were used in the online experiment: CBF-Separated and Pure-CF as a baseline comparison, CF– CBF Separated and CBF Combined – CF as the feature augmentation representatives and the Fusion as the mixed model. The general process of the online experiment is shown in figure 4.8. The arrows show information sent to the next page, cylinders are databases and rectangles are web pages.

According to figure 4.8, when the user comes to the system, she has to consent to participate in the experiment (stage one). After agreeing to participate, the user has to give as input an author's name whose work she is familiar with (stage two). The system then looks up in the authors' database and retrieves all the papers written by this author. With this list in the screen (stage three), the user has to choose one of the papers listed. This paper is, in our experiment, considered as the user profile. In other words, based on this user profile (the paper itself), the system generates recommendations. In stage four, the system shows the recommendations and asks the user to evaluate each one of them. After answering the questions, the user is presented with a new set of questions, now regarding the whole set of recommendations instead of each one individually (stage five). Finally, after submitting these questions, the user is presented with an end page, that just informs the user that the experiment has finished (stage six).

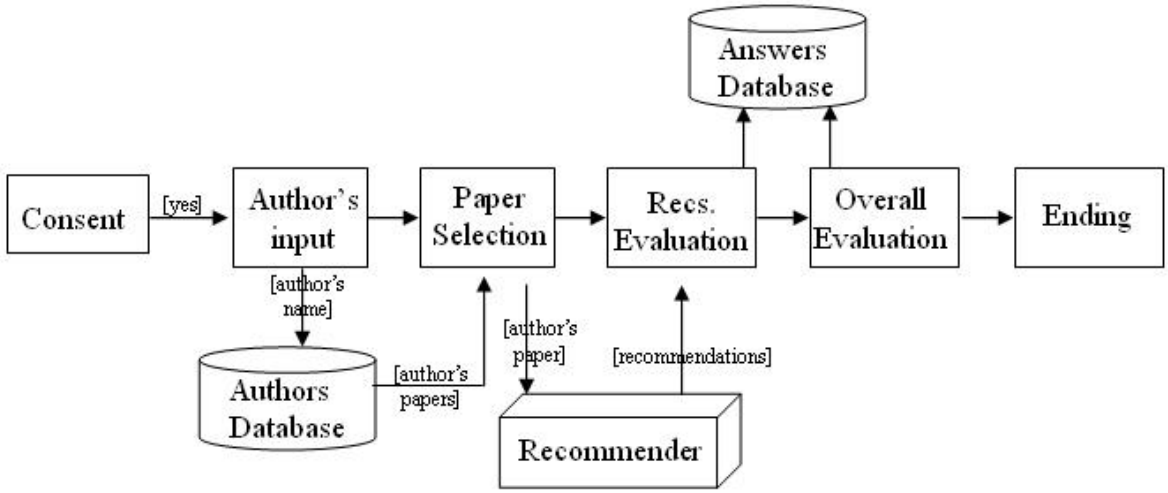

Figure 4.8: Online Experiment – TechLens+

The TechLens^{+} experiment snapshots are shown in appendix C.

4.3.1 Experiment Consent

User task: Consent to participate in the experiment.

This page explains to the user what he/she has to do in the experiment. It briefly explains what are the techniques being studied, what is the experiment

about, who is in charge of the experiment and the tasks he/she has to accomplish in order to successfully finish the experiment. The user could decline or accept to participate in the experiment. A screenshot of the page and the consent text are presented in appendix A.

4.3.2 Author's input

User task: Inform an author's name whose research he/she is familiar with.

In this page the user has to give an author's name. The system searches the author's name in the authors' database and displays his/her papers in the next page. The users are encouraged to type the author's name as they used to appear in their papers (e.g. "John A. Campbell" instead of "J Campbell").

4.3.3 Paper Selection

User task: Select one of the author's papers to get recommendations for.

In this page the user selects one paper of the author he/she has chosen. If the papers presented in this page don't belong to the author informed by the user, the user can choose a different author. The user could also see the paper abstract.

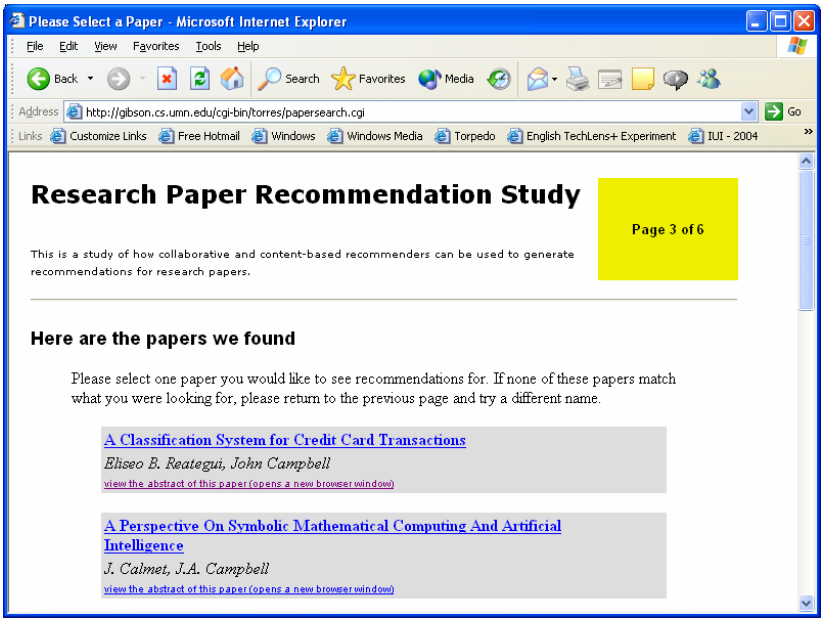

Figure 4.9: Paper Selection

After choosing the paper, the user received recommendations from one of the algorithms. An algorithm was randomly assigned to give recommendations to the user. A screenshot is shown in figure 4.9.

4.3.4 Recommendations' Evaluation

User task: Evaluate every recommendation received.

In this page, the user receives up to five recommendations from the algorithm of the group he/she was assigned. Figure 4.10 shows the questions the user is asked to answer.

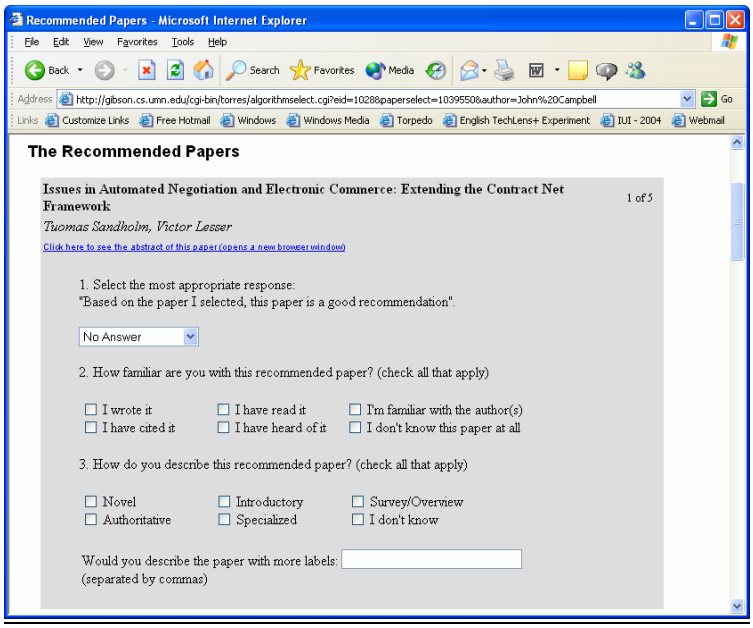

Figure 4.10: Recommendations' Evaluation

The asked questions are listed below:

i. "Based on the paper I selected, this paper is a good recommendation" (Question A)

Options: I strongly agree; I agree; Maybe or unsure; I disagree; I strongly disagree.

Mutually exclusive (radio buttons).

Goal: Measure the quality of every recommendation.

ii. "How familiar are you with this recommended paper?" (Question B)

Multiple choice (checkboxes).

Goal: Measure the user familiarity with the paper.

iii. "How do you describe this recommended paper?" (Question C)

Multiple choice (checkboxes)

Goal: Measure the specificity of the recommender algorithm related to some classes of papers

There was also an empty textbox where users could describe the recommended papers with more opinions.

4.3.5 Overall Recommendations' Evaluation

User task: Evaluate the whole set of the received recommendations.

In this page, the user is asked to answer four questions, regarding the overall set of recommendations and the user him/herself. The questions were:

iv. "For what applications would you be interested in using a research paper recommender system like this one?" (Question D)

Multiple choice (checkboxes). There's also a free-form label for users to enter with more possible applications.

Goal: Find for which applications the algorithm is more suitable

v. "Do you think that the overall set of recommendations were useful?" (Question E)

Mutually exclusive (radio buttons).

Goal: Measure the user perception about the quality of the whole set of recommendations.

vi. "Which of the following attributes do you think a recommender system like this one should take into account when generating recommendations?" (Question F)

Multiple choice (checkboxes). There's also a free-form label for users to enter with more possible attributes.

Goal: Prospect from the user which other attributes might be explored in a research paper recommender system.

vii. "How would you describe yourself?" (Question G)

Mutually exclusive (radio buttons).

Goal: Evaluate how different users perceive the system.

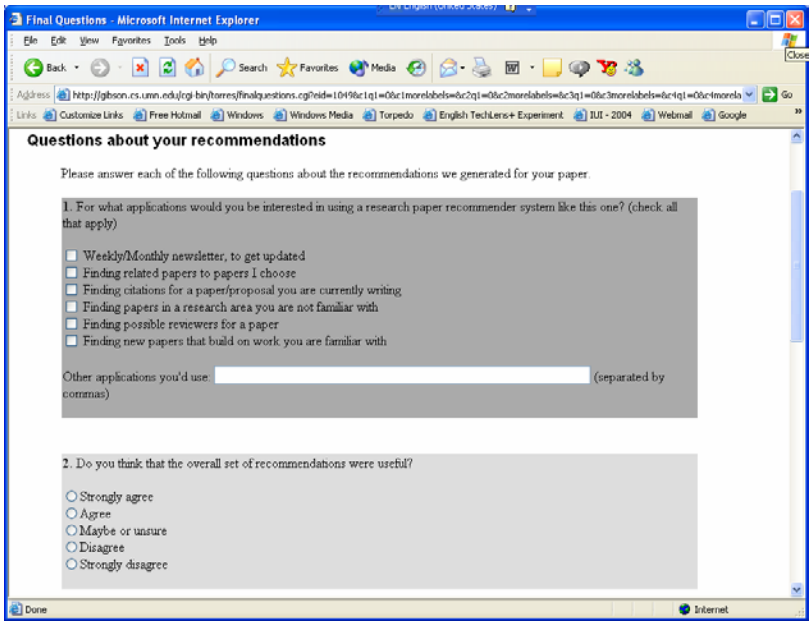

Figure 4.11: Overall Evaluation - A

For some users is possible that there is an overlap in their descriptions (e.g. a user can be a professor and a researcher at the same time). It is up to the user how he/she describes him/herself. At the end of the page there is also a box for additional comments, regarding the whole experiment. A screenshot of the page is shown in figures 4.11 and 4.12.

4.3.6 Online Experiment Ending

In this page, the system just informs the user that he/she has finished the experiment. At this point, all the information provided by the user has been saved.

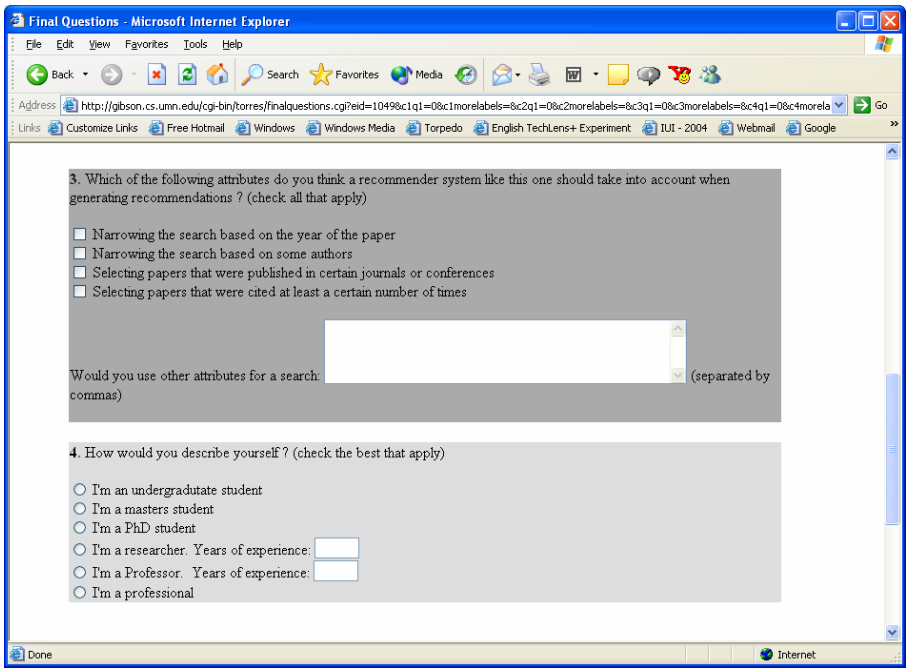

Figure 4.12: Overall Evaluation - B

4.3.7 Online Experiment's Results

We can summarize the data collected in the TechLens$^+$ system as shown in table 4.2. Our experiment assessed not only the quality of the recommendations, but also who were evaluating them and their experience. These evaluations allowed us to verify that different algorithms are more suitable for different kinds of users, and for users with different levels of experience. For instance, an undergraduate student would be more interested in an introductory paper than a researcher or a professor would be. Users ranged from undergraduate students to professors and the invitations were sent during the experiment, on a week basis. After the launch of the TechLens$^+$ Portuguese version, invitations to Brazilian people were sent in Portuguese. Otherwise, they were sent in English to everyone else.

Table 4.2: TechLens+ Collected Data

Objective	Metric	Indicators	Question number
Quality of recommendations	User satisfaction	Agreement of user that is a good recomm.	A, E
Background of user	Familiarity with paper subject	Previous contact with the paper	B
	Class of user	Experience in research activity	G
Type of algorithm related with some classes of applications	User answer	Types of applications pointed by the user	D
Type of algorithm related with some classes of papers	Distribution of the papers in the same class	Classes of the papers pointed by the user	C
Type of algorithm related with attributes used on recommendations	User answer	Types of attributes pointed by the user	F

During the 32-day experimental run, 110 subjects participated in the experiment: 33 from the United States, 43 from Brazil and 34 from other countries[6]. On average, subjects spent 20 minutes answering our questions. Also, 20 subjects were Masters students, 33 were Ph.D. students, 27 were researchers and 23 were professors. Undergraduate students and professionals represented 6 subjects and were not separately analyzed. The number of users that received recommendations per each algorithm is shown in table 4.3.

Table 4.3: Users per Algorithm

Algorithm	Number of users
CBF Separated	14
Pure-CF	28
CF-CBF Separated	25
CBF Combined – CF	18
Fusion	25

In order to evaluate user's satisfaction about their recommendations, we categorized their answers. The options strongly agree and agree options are considered as "satisfied" and strongly disagree and disagree are considered as "dissatisfied", regarding the quality of the recommendations. Non-committal answers (e.g. unsure) were ignored. As table 4.4 shows, subjects were satisfied both for each individual recommendation and overall. The interpretation of the table is as follows: 46% of the recommendations made users satisfied about them and 62% of the users were satisfied about the overall set of recommendations. Figure 4.13 shows user satisfaction for each algorithm. CBF-Separated, Fusion, and CBF Combined-CF scored higher than Pure-CF and CF-CBF Separated.

Table 4.4: Users' Satisfaction about Recommendations

	Individual Recommendations	Overall Set of Recommendations
Satisfied	46%	62%
Dissatisfied	21%	19%

To evaluate user's familiarity about the papers, we broke down our analysis into three groups: a user is considered *very familiar* if he/she cited or read the paper, *familiar* if he/she has heard about the paper or is familiar with the authors, and *unfamiliar* if he/she doesn't know the recommendations at all. Of all the recommendations, 27% were very familiar to the users, 34% were familiar, and 36% of the papers were unfamiliar. Only 2% of the recommendations received were written by the users who were evaluating them. Recommendation satisfaction also varied by user type with 75% of the masters students, 61% of the PhD students, 67% of the researchers, and only 52% of the professors saying they were satisfied with their recommendations.

[6] More than 10 countries, though not worth analyzing

- Paper Class Analysis

As we asked users to describe papers with five options (see 4.3.4) and that every user received recommendations from only one algorithm, we could find out which algorithms are better for recommending different kinds of papers. Hence, table 4.5 summarizes the best and worst algorithms for each class of paper. In terms of statistical significance, Pure-CF and Fusion are better than CF-CBF Separated for recommending novel and authoritative papers.[7]

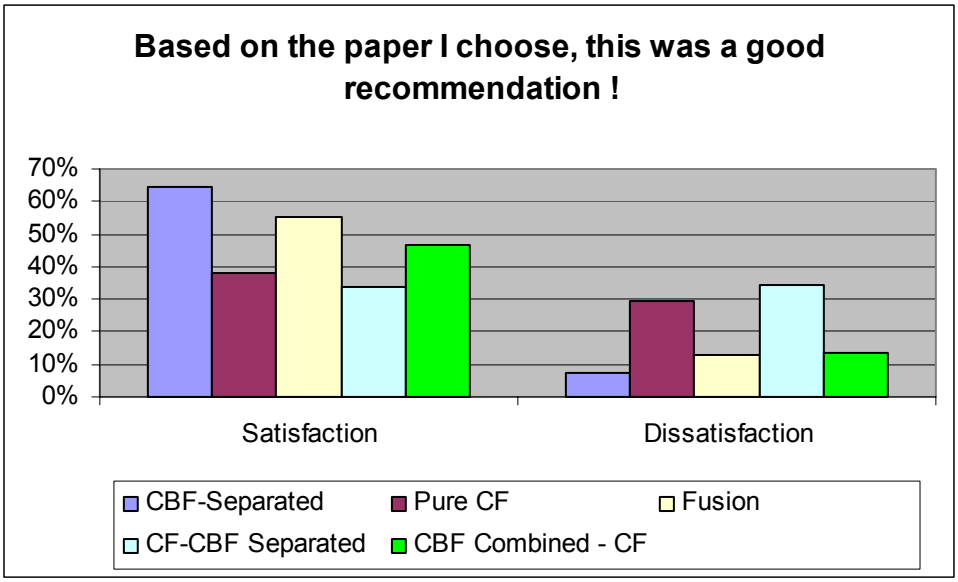

Figure 4.13: User Satisfaction by Algorithm

For introductory papers, CBF-Separated and CF-CBF Separated are better than Pure-CF. Finally, CBF-Separated is better than Pure-CF and Pure-CF was worse than all of the other algorithms to recommend survey papers ($p < 0.1$).

Table 4.5: Recommender Algorithms by Paper Class

Class of Papers	Best Algorithms	Worst Algorithms
Novel	Pure-CF and Fusion	CBF-Separated
Authoritative	Pure-CF and Fusion	CF-CBF Separated
Introductory	CBF-Separated and CF-CBF Separated	Pure-CF
Survey/Overview	CBF-Separated	Pure-CF

We also surveyed the users about what kinds of application they would be willing to use a research paper recommender: 91% of the users answered “find related papers to papers I chose”; 72% answered “finding citations for a paper/proposal you are currently writing”; 64% answered “finding papers in a research area you are not familiar with”; and 39% would be willing to receive a weekly/monthly newsletter with paper recommendations.

- Cross-country Analysis

[7] Unless otherwise noted, significance tests are at $p < 0.05$

Approximately 2/3 of the users came either from the United States or Brazil. This was expected because a large audience that participated in our experiment came from the universities, and all universities we sent invitations to were from Brazil or United States. A user is considered from one of these countries based on where the user was physically located when he/she accessed the experiment (via IP address). The breakdown of the subject population is shown in Table 4.6b.

User satisfaction with individual recommendations is similar, with 50% satisfaction by Americans and 49% by Brazilians. Dissatisfaction is similar too: Americans at 15% and Brazilians at 17%. On the other hand, satisfaction with the overall set of recommendations varied greatly. Americans were satisfied with 42% and not satisfied with 33% of the recommendations, while Brazilians were satisfied with 70% and not satisfied with 12% of the recommendations.

Table 4.6: Distribution of Users

Brazil Eng.	Brazil Port.	Type of User	Total Brazil	Total USA
3	12	Masters Students	15	4
3	7	PhD Students	10	13
0	6	Researchers	6	8
5	5	Professors	10	5

(a) (b)

There were also strong differences in familiarity. Americans were more familiar with the recommendations, with 31% very familiar, 41% familiar and 24% unfamiliar. Brazilians, on the other hand, were 24% very familiar, 31% familiar and 44% unfamiliar with the recommendations. Thus, Americans were more familiar with the recommendations and less satisfied than Brazilians.

- Cross-Language Analysis

The Portuguese version of TechLens$^+$ started 6 days after the English version. During this time, 12 Brazilian users participated in the English Version of the experiment. After the launch of the Portuguese version, Brazilian users preferred to participate in this version. We then divided the Brazilians into two groups: those that participated in the English and those that participated in the Portuguese version. This population distribution is shown in Table 4.6a.

Overall recommendation quality showed stronger differences: Brazilians were satisfied with 42% and dissatisfied with 33% of the recommendations in the English version, while in Portuguese, they were satisfied with 81% and dissatisfied with only 3% of the recommendations.

Finally, in English, Brazilians were 11% very familiar, 32% familiar and 57% unfamiliar. While in Portuguese, they were 29% very familiar, 31% familiar and 40% unfamiliar with the recommendations. Thus, Brazilians in the Portuguese experiment were more familiar than Brazilians in the English experiment. Although not statistically significant, trends in the data show that Brazilians in the Portuguese experiment searched more for Brazilian authors than in the English experiment.

5 ANALYSIS AND DISCUSSION

This research focused on finding out issues on recommending research papers to users, including development of new algorithms. In order to do so, we have performed two experimental evaluations, one offline and one online. The methodological process used, selecting best algorithms and then evaluating them with users led us to draw analysis and discussions in many dimensions. A total of ten algorithms were tested offline and five were selected to go online. The criteria used to select the bests offline was based on its ability to recommend good recommendations soon (top-10 and top-1) and based on the number of recommendations it could generate ("all").

CBF-Separated had the highest user satisfaction among all algorithms. We believe that it happened because it generated similar papers to the one the user has chosen, giving them the feeling of better recommendations.

Pure-CF was good at generating novel and authoritative papers. This is inherent to the technique: novelty is obtained by the serendipitous behavior and authoritativeness is obtained by the citation web analysis because often cited papers are more likely to be recommended. On the other hand, CBF algorithms were good at finding introductory and survey papers. These papers generally have a large content overlap with the papers they introduce or overview, thus CBF performed well.

Fusion also had a high user satisfaction and its recommendations were more balanced according to the characteristics of both CF and CBF. Fusion recommendations' quality was expected because it had in the offline experiment the highest top-1 hit-percentage (28%), almost two times the second highest Pure-CF (15%). Despites finding statistical significance in the data we gathered, in the long run we expect Fusion to perform better than the others. Additionally, because the techniques run in parallel, Fusion can automatically incorporates any enhancement to each technique individually. Finally, as figure 4.13 shows, TechLens^{+} could though successfully achieve its goal "A", evaluating the quality of the recommendations given by the algorithms.

Users also gave feedback about the system. One said: "I was looking for papers that would help me writing a compiler without writing code generators for many different processors". The fact that the user found one good recommendation out of five made him/her "happy" about the performance of the system, particularly because he/she was looking for something very specific. Therefore, one good recommendation out of five is a very good result in real systems, and 85% of our users got at least one good recommendation. Other users also gave us feedback about the usefulness of the system in recommending novel papers: "The recommendations

were useful as they represent papers I haven't seen before. However papers I thought would appear did not".

Regarding the goal "B" of the TechLens^{+} system, our cross-country analysis showed that *there are no strong cultural differences in receiving research paper recommendations*. In addition, our analysis reinforced the results that the level of experience influences the users' satisfaction (H3). Brazilian users had a higher percentage of masters' students than American users had. Consequently, Brazilian users were less familiar about the papers than American users. Therefore, the recommendations given to Brazilian users made them more satisfied.

Our analysis also showed strong language differences. Brazilians in the Portuguese experiment were both more familiar and more satisfied with the recommendations they received. We hypothesize that because most of what is in the Internet is written in English, Brazilian users might be more satisfied being invited to participate in a Portuguese experiment. This suggests that research paper Recommender Systems interface should be localized to the user's native language, reducing the users' burden of finding good research papers. This is independent of the language of the papers themselves, because most of them were written in English, and Brazilians were happy either way.

The poor performance of Denser-CF was a surprising result because we thought that the denser input used in this algorithm should improve the quality of recommendations. Because our dataset has an average of 14 connections per paper, we could assume that every paper cites an average other 7 papers and is cited by other 7. Denser-CF then is augmenting in average the set of citations of a given paper by 49 citations, which might be not closely enough related to the active paper, generating only noise.

Another advantage of the hybrid approaches developed in this research is that they could recommend a brand new research paper. This is due to the presence of CBF. Using only CF, a paper would be recommended only if some other paper has cited it. Because of the dynamism of the research paper domain, with thousands of new papers published every year, an effective algorithm has to be able to recommend just-published papers.

6 CONCLUSIONS AND FUTURE WORK

In this research we described, implemented and tested different techniques for combining content-based and Collaborative Filtering recommender algorithms for recommending research papers. We verified our results through both offline and online experiments. It is important to stress that 85% of the users said they received at least one good recommendation, and that 65% said they received at least two good ones. Moreover, although the experiments conducted on this research have focused on Computer Science research papers, the algorithms developed in this research can be easily adapted to any domain of Science, as long the text and citations of the papers are available.

This research presented many unique approaches to recommend research papers and, besides the results found, it leaves two other important contributions: (i) a set of algorithms that can be used for recommending research paper in many domains of science; and (ii) an online experiment to assess users' perceptions about the recommendations. Of particular note is CBF-Separated, which despite not being hybrid, had the highest user satisfaction and its internal characteristics allow it to generate serendipitous recommendations as well.

Returning to our hypotheses, we found that CF-CBF hybrid recommender algorithms can generate research paper recommendations equal to or better than CF or CBF alone (H1). Furthermore, all hybrid algorithms had 100% coverage online, which is a significant advantage of a hybrid algorithm. In contrast, most of the feature augmentation algorithms we tested did not perform well online. We believe this is due to the sequential nature of these algorithms: the second module is only able to make recommendations seeded by the results of the first module. In general, we believe sequential hybrid recommendation algorithms will not perform well because pure recommender algorithms are not designed to receive input from another recommender algorithm. Thus, future research should focus on nonlinear hybrid models.

Still regarding H1, the users' preference for a non-hybrid algorithm in this research might have been influenced by its sample size, with only 25 users evaluating the algorithm fusion. Further analysis is necessary to explore what happened. However, although the algorithm with highest user satisfaction wasn't hybrid, we believe that in the long run CF-CBF hybrid algorithms tend to perform better.

Our online results showed that different algorithms should be used for recommending different kinds of papers (H2), reinforcing results found in previous work (McNee;Albert et al. 2002).

Regarding H3, we did find that the level of experience influences users' satisfactions about recommendations. Students (masters or PhD) were happier about the received recommendations than professionals (researchers or professors). It is also important to point out that our cross-country analysis did not show any strong differences in how Americans and Brazilians perceive recommendations, except in the overall analysis where Brazilians were a little happier. However, our cross-language analysis showed strong differences in users' perceptions.

Therefore, our results lead us to a vision of a completely personalized or 'tailored' recommender system. Our results suggest such a system can be tailored in three ways. First, it can tailor recommender algorithms for particular user tasks using Table 4.5 as a guide. For example, suppose that the task of "finding related work" could be solved by recommending novel and authoritative papers. Then a system that wanted to support this task should use Pure-CF and Fusion to generate paper recommendations. Second, our cross-language analysis suggests that a system should tailor itself to the user's native language. Finally, our online experiment and our cross-country and user-experience analysis suggest that a system should tailor the recommendations it displays based on the level of experience a user has. More research is needed both to effectively determine how to perform these tailorings, and how to dynamically update a user interface to display the recommendations in a way that won't confuse users.

Our offline experiments suggested that the high distribution of content in larger datasets limits the ability of CBF recommendation. This could be due to the trimming we did, eliminating citations for which we didn't have the text. This requires further research, along with studies about how the density of a dataset influences the quality of the recommendations. In the future, most papers will be available online, probably improving the quality of the recommendations.

In addition the dataset density research, it would also be interesting to explore how the recommendations in a dataset are when the papers have a few number of citations. This is important because journal papers have much more scientific value than conference papers and usually they have less number of citations.

In the future, it would also be interesting to investigate algorithm differences in recommending recent compared to older research papers. We believe this leads to the possibility of recommending "research paths" to users. Given a query of a research area and knowledge of what a user has already read, a recommender could generate a display of how this area has evolved over time and produce an ordered list of "must-read" papers in that field. We believe this is an important area to look into, not only for educational purposes, but because over 69% of our subjects said they would like Recommender Systems to help them find papers that built on previously known research.

REFERENCES

ACM Digital Library. Available at: www.acm.org. Accessed in: April 2003.

AMAZON. Available at: www.amazon.com. Accessed in: May 2003.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**.New York: ACM, 1999.

BALABANOVIC, M.; SHOHAM, Y. Fab: Content-Based, Collaborative Recommendation. **Communications of the ACM**, New York, v. 40, n. 3, p. 66-72, Mar. 1997.

BASU, C. et al. Recommendation as Classification: using Social and Content-based Information in Recommendation. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 15, 1998, Madison, Wisconsin - USA. **Proceedings...** Menlo Park: AAAI Press, 1998.

BILLSUS, D.; PAZZANI, M. J. Learning Collaborative Information Filters. WORKSHOP ON RECOMMENDER SYSTEMS, 1998, Madison, Wisconsin - USA. **Procedings…** [S.l.]: AAAI Press, 1998.

BOLLACKER, K. et al. A System for Automatic Personalized Tracking of Scientific Literature on the Web. In: ACM CONFERENCE ON DIGITAL LIBRARIES, 4, 1999, Berkeley, CA. **Procedings…** [S.l. : s.n.], 1999.

BREESE, J. et al. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 1998, Madison, Wisconsin. **Proceedings...** [S.l. : s.n.], 1998.

BURKE, R. Hybrid Recommender Systems: Survey and Experiments. **User Modeling and User-adapted Interaction**, [S.l.], v. 12, n. 4, p. 331-370, Nov 2002.

CLAYPOOL, M. et al. Combining Content-Based and Collaborative Filters in an Online Newspaper. In: ACM SIGIR WORKSHOP ON RECOMMENDER SYSTEMS, 1999, Berkeley - CA. **Proceedings...** [S.l. : s.n.], 1999.

CLAYPOOL, M. et al. Implicit interest indicators. International Conference on Intelligent User Interfaces, 2001, Santa Fe - United States. **Proceedings...** [S.l. : s.n.], 2001.

COTTER, P.; SMYTH, B. PTV: Intelligent Personalised TV Guides. In: INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE, 12, 2000, Austin, Texas. **Proceedings...** [S.l. : s.n.], 2000.

EBIZSEARCH. Available at: http://gunther.smeal.psu.edu/. Accessed in: September 2003.

NATIONAL SCIENCE FOUNDATION. Academic Research and Development. Available at: http://www.nsf.gov/sbe/srs/seind02/c5/c5s3.htm. Accessed in: September 2003.

GOLDBERG, D. et al. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, New York, v. 35, n. 12, p. 61-70, December 1992.

GOOGLE. Available at: www.google.com. Accessed in: May 2003.

HERLOCKER, J. et al. An Algorithmic Framework for Performing Collaborative Filtering. In: ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 1999, Berkeley, CA. **Proceedings...** [S.l. : s.n.], 1999.

HERLOCKER, J. L. **Understanding and Improving Automated Collaborative Filtering Systems**. 2000. Phd Thesis. Department of Computer Science and Engineering, University of Minnesota, Minneapolis - MN - USA.

HILL, W. et al. Recommending and Evaluating Choices in a Virtual Community of Use. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1995, Denver - CO - USA. **Proceedings...** New York: ACM Press, 1995.

KAMBA, T. et al. The Krakatoa Chronicle: An Interactive Personalized Newspaper on the Web. INTERNATIONAL WORLD WIDE WEB CONFERENCE, 4, 1995, Boston. **Proceedings...** O'Reilly & Associates, 1995.

ISI Web of Knowledge. Available at: http://www.isinet.com/. Accessed in: April 2003.

KROON, H. C. M. D. et al. Improving Learning Accuracy in Information Filtering. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 13, 1996, Bari, Italy. **Proceedings...** [S.l. : s.n.], 1996.

LANL, e-Print Archive. Available at: http://arxiv.org/. Accessed in: April 2003.

LAWRENCE, S. et al. Autonomous Citation Matching. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 3, 1999, Seattle, Washington - USA. **Proceedings...** [S.l. : s.n.], 1999.

LIMA, P. S. R.; PIMENTA, M. **Personalização de Interfaces Web para Sites Institucionais com Base em Perfis de Usuários**. 2002. Dissertação (Mestrado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre - RS.

MALTZ, D.; EHRLICH, K. Pointing the way: Active collaborative filtering. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1995, New York. **Proceedings...** New York: ACM, 1995.

MCCALLUM, A. K. **Bow**: A toolkit for statistical language modeling, text retrieval, classification and clustering. Available at: http://www.cs.cmu.edu/~mccallum/bow. Accessed in: April 2003

MCGINTY, L.; SMYTH, B. Collaborative Case-Based Reasoning: Applications in Personalised Route Planning. In: INTERNATIONAL CONFERENCE ON CASE-BASED REASONING, 4, 2001, Berlin, Germany. **Proceedings...** Berlin: Springer Verlag, 2001.

MCNEE, S. et al. On the Recommending of Citations for Research Papers. In: COMPUTER SUPPORTED COOPERATIVE WORK CONFERENCE, 2002, New Orleans, Louisiana - USA. **Proceedings...** New York: ACM, 2002.

MELVILLE, P., et al. Content-Boosted Collaborative Filtering. ACM WORKSHOP ON RECOMMENDER SYSTEMS, 2001, New Orleans - LA. **Procedings...** 2001.

MILLER, B. **Towards a Personal Recommender System**. 2003. PhD Thesis. Department of Computer Science and Engineering, University of Minnesota, Minneapolis.

NEW-ZEALAND Digital Library. Available at: http://www.sadl.uleth.ca/nz/cgi-bin/library. Accessed in: April 2003.

PAZZANI, M. J. A Framework for Collaborative, Content-Based and Demographic Filtering. **Artificial Intelligent Review**, [S.l.], v. 13, n. 5-6, p. 393-408, 1999.

PENN State CiteSeer. Available at: http://citeseer.ist.psu.edu/. Accessed in: September 2003.

PORTER, M. F. An Algorithm for Suffix Stripping. **Program**, [S.l.], v. 14, n. 3, p. 130-137, July 1980.

RASHID, A. M. et al. Getting to Know You: Learning New User Preferences in Recommender Systems. In: INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES, 2002, San Francisco, CA - USA. **Proceedings...** [S.l. : s.n.], 2002.

REATEGUI, E.; CAMPBELL, J. A. The Role of Personified Agent in Recommendation Delivery. ACM WORKSHOP ON RECOMMENDER SYSTEMS, 2001, New Orleans - USA. **Proceedings...** [S.l.: s.n.], 2001.

REATEGUI, E., TORRES, R. et al. Personalizing with Recommendation Frames. In: ACM WORKSHOP ON RECOMMENDER SYSTEMS, 2001, New Orleans - USA. **Proceedings...** [S.l.: s.n.], 2001.

RESNICK, P. et al. GroupLens: An open architecture for collaborative filtering of netnews. Computer Supported Collaborative Work Conference, 1994, Chapel Hill, North Carolina - USA. **Proceedings...** [S.l.: s.n.], 1994.

RIEDL, J.; KONSTAN, J. **Word of Mouse**:The Marketing Power of Collaborative Filtering. New York: Warner Books, 2002.

SALTON, G.; BUCKLEY, C. Term weighting approaches in automatic text retrieval. **Information Processing and Management**, [S.l.], v. 24, n. 5, p. 513-523, 1988.

SARWAR, B. et al. Analysis of Recommender Algorithms for E-Commerce. In: ACM E-COMMERCE CONFERENCE, 2000 **Proceedings...** [S.l.: s.n.], 2000.

SARWAR, B. et al. Item-based Collaborative Filtering Recommendation Algorithms. WORLD WIDE WEB CONFERENCE, 2001, Hong Kong, China. **Proceedings...** [S.l.: s.n.], 2001.

SCHAFER, B. et al. Recommender Systems in E-Commerce. In: ACM CONFERENCE ON ELETRONIC COMMERCE, 1999, Denver, Colorado - USA. **Proceedings...** [S.l.: s.n.], 1999.

SHARDANAND, U.; MAES, P. Social information Filtering: Algorithms for automating "word of mouth". In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1995, New York. **Human Factors in Computing Systems.** New York: ACM, 1995.

SUGGEST Top-N Recommendation Engine. Available at: http://www.cs.umn.edu/~karypis/suggest/. Accessed in: April 2003.

TERVEEN, L. et al. PHOAKS: A system for Sharing Recommendations. **Communications of the ACM,** New York, v. 40, n. 3, p. 59-62, 1997.

WOODRUFF, A. et al. Enhancing a Digital Book with a Reading Recommender. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2000, The Haghe, The Netherlands. **Proceedings...** [S.l. : s.n.], 2000.

APPENDIX A CONSENT FORM

Collaborative filtering has been shown to be useful in making personalized recommendations for users in a wide variety of environments. It has been used successfully in such domains as movies, music, and jokes. Content-based Filtering has been used to find information and generate recommendations in information retrieval systems for many years. This experiment explores how both Collaborative Filtering and Content-based Filtering can be used to generate recommendations in the domain of computer science research papers.

This research is being conducted by Sean M. McNee (University of Minnesota - USA) and Roberto D. Torres Jr. (UFRGS - Brazil) from the GroupLens Research Project from the Department of Computer Science and Engineering at the University of Minnesota-Twin Cities under the supervision of Prof. John Riedl and Prof. Joseph A. Konstan (University of Minnesota - USA) and Mara Abel (UFRGS - Brazil). Questions or concerns about this study can be emailed to torres@cs.umn.edu.

In order to participate in this study, you must be familiar with a paper that has been cataloged and indexed by Research Index, an online scientific literature digital library. We ask that you read this form and ask any questions you may have before agreeing to be in the study. If you agree to be in the study, we will ask you to do the following things:

1. Provide us with an author's name, so that we can look up paper written by that author.

2. Select a paper from a list we will generate.

3. Evaluate, on multiple scales, five citations that will be recommended for the paper you selected.

4. Answer a few questions about the overall quality and usefulness of the recommended citations.

The study is six pages in length, including this page, and it should take between five and fifteen minutes to complete.

You will be assigned to an experimental group. Each group will receive recommendations generated from a different algorithm. Some algorithms may not use Collaborative Filtering to generate recommendations.

The study has no known significant risks. Please be aware that the recommendations may not be very precise due to insufficient data or mathematical

modeling. It is also possible that we cannot generate recommendations for any specific paper due to limitations in our testing dataset.

By participating in this study, you will have the opportunity to try new cutting edge research technology that we hope will be able to help you filter information faster and more efficiently. You will be among the first people to experience this technology. Your privacy will be protected. Any data that we collect will be fully anonymized before it is released to the public or described in any publication.

Your decision whether or not to participate will not affect your current or future relations with Research Index or the University of Minnesota. If you decide to participate, you are free to withdraw at any time without affecting those relationships.

If you have any questions or concerns regarding the study and would like to talk to someone other than the researcher(s), contact Research Subjects' Advocate line, D528 Mayo, 420 Delaware Street S.E., Minneapolis, Minnesota 55455; telephone (612) 625-1650.

APPENDIX B EXTENDED SUMMARY (IN PORTUGUESE)

Título: Combinando Filtragem Colaborativa e Filtragem Baseada em Conteúdo para Recomendar Artigos Científicos

1. Introdução

Hoje em dia há uma sobrecarga de artigos científicos disponíveis que estudantes e pesquisadores não conseguem administrar. De acordo com a National Science Foundation dos Estados Unidos, mais de 530.000 artigos foram publicados em mais de 1.900 revistas científicas, apenas no ano de 1999. Além disso, a taxa de aumento do número de artigos publicados a cada ano cresce em 1% desde 1986. Se essa tendência continuar, mais de 10 milhões de novos artigos serão publicados nos próximos 20 anos. Com o surgimento da Internet, essa situação tornou-se ainda pior, tendo em vista que os artigos tornam-se prontamente disponíveis em diversos sites, como em bibliotecas digitais.

Sistemas de Recomendação (SR) foram desenvolvidos na década passada com o objetivo de reduzir a sobrecarga de informações existente e têm sido aplicados nos mais diversos domínios, como sistemas de notícias (Resnick;Iacovou et al. 1994), filmes (Hill;Stead et al. 1995; Herlocker;Konstan et al. 1999), CDs (Shardanand;Maes 1995), guias de televisão (Cotter;Smyth 2000). Pesquisadores têm, para cada domínio, aplicado um grande conjunto de técnicas, cada qual com suas forças e fraquezas.

Para lidar com o elevado número de artigos existentes, foram construídas bibliotecas digitais (LANL; NZDL) e sistemas como CiteSeer (Bollacker;Lawrence et al. 1999), que armazenam artigos num repositório centralizado e permite que os usuários realizem consultas parametrizadas.

O elevado número de artigos disponíveis on-line e a quantidade de novos artigos produzidos por ano tornam esse domínio um bom alvo para Sistemas de Recomendação. Além disso, artigos possuem duas propriedades particularmente

interessantes: o *texto* dos artigos em si e a *cadeia de citações*, que liga artigos a outros artigos relevantes. Essas propriedades podem ser completamente exploradas por técnicas de SR, tanto analisando o texto dos artigos como encontrando artigos relevantes baseado na ligação com outros artigos.

Portanto, o objetivo dessa dissertação é analisar diversas questões sobre recomendação de artigos científicos. Para isso, a abordagem utilizada nessa pesquisa é a combinação de duas técnicas bem conhecidas de SR, chamadas Filtragem Colaborativa (FC) e Filtragem Baseada em Conteúdo (FBC). É importante salientar que esse trabalho é baseado em um trabalho anterior, que conseguiu com sucesso aplicar FC em recomendação de artigos (McNee;Albert et al. 2002), mas não explorou o potencial das duas técnicas juntamente.

Essa pesquisa então define três hipóteses, que serão validadas ou não de acordo com os resultados obtidos em nossos experimentos *off-line* e *on-line*.

H1: Filtragem Colaborativa e Filtragem Baseada em Conteúdo podem ser combinadas para gerar recomendação de artigos científicos

H2: Diferentes algoritmos são mais adequados para recomendar diferentes tipos de artigos.

H3: Usuários com diferentes níveis de experiência percebem recomendações de artigos científicos diferentemente.

A primeira hipótese é baseada no fato de que as duas técnicas podem, isoladamente, ser utilizadas para recomendação de artigos. Entretanto, FC e FBC possuem características complementares que devem ser eliminadas em uma abordagem híbrida. Tendo em vista que existem diferentes tipos de artigos, como introdutórios e *surveys*, a segunda hipótese sugere que há algoritmos mais adequados para recomendar cada tipo de artigo. Por último, a terceira hipótese busca explorar em como a diferença no nível de experiências dos usuários, que por exemplo podem ser estudantes inexperientes ou professores com anos de experiência, influenciam a recomendação de artigos.

Além dessas hipóteses, essa pesquisa tem por objetivo explorar como usuários de diferentes países percebem recomendações de artigos. Mais ainda, investigamos diferenças em idiomas. Para isso, conduzimos um experimento *on-line* com usuários de diferentes países e em diferentes idiomas.

2. Validação das Hipóteses

Para a validação de nossas hipóteses, criamos versões autônomas e híbridas de algoritmos e os avaliamos através de experimentos *off-line* (sem a participação do usuário) em uma base de dados de 102,295 artigos, e através de um experimento *on-line* com 110 usuários.

EXPERIMENTOS *OFF-LINE*

Os experimentos *off-line* testam se os algoritmos são úteis para sugerir artigos relevantes para um determinado artigo (artigo corrente). Para cada artigo é retirada aleatoriamente uma citação e o algoritmo deve tentar recomendar essa citação. Para isso foram criadas conjuntos de dados de treinamento e teste, numa razão de 90% e 10%. Foram gerados 10 diferentes arquivos de treinamento e teste.

Esse método de avaliação possui algumas limitações. Os algoritmos de recomendação podem recomendar artigos que não existiam na época em que o artigo corrente foi publicado. Para lidar com isso, artigos que tenham ano de publicação maior que o ano de publicação do artigo corrente são descartados. Além disso, é importante salientar que os algoritmos podem recomendar artigos muito similares ou até melhores que o artigo removido das citações do artigo corrente, dando a impressão de um desempenho ruim do algoritmo. Apesar de isso ser uma possibilidade, esperamos que a citação removida seja a recomendada.

Os algoritmos foram testados em duas bases de dados, chamadas *grande* e *pequena*. A base de dados grande possui 102.295 artigos e a base pequena possui 1.173 artigos. Para avaliar o desempenho dos algoritmos, definimos o "percentual de acertos", que é o percentual das vezes que um algoritmo recomenda a citação removida. Nós também medimos a posição em que a citação removida foi recomendada. Além disso, tendo em vista que a posição da recomendação é importante para o usuário, dividimos nossa análise em faixas de posições. Por exemplo, é melhor que a citação removida seja encontrada entre as 10 primeiras (top-10) recomendações do que entre as 40 primeiras (top-40). A tabela 1 mostra as faixas consideradas.

Tabela 1: Análise de posições

Posição da citação	Faixa
1	top-1
1-10	top-10
1-20	top-20
1-30	top-30
1-40	top-40
1-N	All

Dez diferentes algoritmos foram desenvolvidos, sendo cinco não-híbridos (autônomos), utilizados para comparação, e cinco híbridos. Dos cinco autônomos, três utilizam apenas FBC (Pure-CBF, CBF-Separated e CBF-Combined), dois utilizam apenas FC (Pure-CF e Denser-CF). Dos cinco híbridos, quatro utilizam FC e FBC em modo seqüencial, utilizando as recomendações de uma técnica como entrada de dados para a segunda técnica (CF-CBF Separated, CF-CBF Combined, CBF Separated-CF, CBF Combined-CF) e apenas um utiliza as duas técnicas em paralelo, com as recomendações vindo ao mesmo tempo das duas listas (Fusion).

Dos dez algoritmos desenvolvidos e testados, cinco são utilizados no experimento on-line. Para selecionar os melhores nos experimentos off-line, consideram-se os resultados na base *grande* e são utilizados os seguintes critérios: serão selecionados os que obterem o melhor resultado na análise top-10 e "all". Um terceiro critério é usado quando o top-10 de um algoritmo é melhor mas o "all" de outro algoritmo é melhor. Nesse, caso, top-1 decide qual é o melhor.

Para analisar ao menos um algoritmo de cada tipo, esses os critérios acima definidos serão aplicados nas seguintes classes de algoritmos: autônomos que utilizam FBC, autônomos que utilizam FC, híbridos em seqüência começando com FC e FBC, e híbridos em paralelo. Portanto, de acordo com esses critérios, os melhores algoritmos foram: CBF-Separated, Pure-CF, CF-CBF Separated, CBF Combined – CF e Fusion.

Os algoritmos tiveram, em média, 33% de acerto na análise top-10. Destaque para o algoritmo Pure-CF, que teve 43%, sendo o melhor dentre os cinco. Considerando a faixa "all", a média foi de 65%, sendo Fusion o melhor com 78%. Fusion merece ainda um destaque especial, tendo em vista seu resultado top-1 que foi de 28%, quase o dobro do segundo melhor algoritmo, Pure-CF, que foi de 15%.

CBF-Separated era esperado que tivesse um bom desempenho tendo em vista que aumenta o espaço de busca, buscando por documentos similares não somente ao artigo corrente, mas também às suas citações. Denser-CF era esperado ter um resultado melhor pois utiliza uma matriz mais densa e a esparsidade é um dos problemas de filtragem colaborativa. Entretanto, Pure-CF teve um melhor desempenho em todas as faixas de análise. Fusion, que obteve o maior percentual de acertos, também era esperado ter um bom resultado visto que as melhores recomendações de cada técnica individualmente são colocadas em uma mesma lista, ordenadas por suas posições nas listas de recomendações iniciais.

Também é importante salientar que todos os algoritmos tiveram cobertura de 100%, ou seja, puderam gerar recomendações para todos os artigos. Isso é devido principalmente à presença do componente de FBC nas técnicas híbridas.

EXPERIMENTOS *ON-LINE*

Apesar de os experimentos *off-line* serem bons indicativos da qualidade dos algoritmos, somente usuários poderiam nos dar uma real percepção da qualidade dos algoritmos. Para isso, desenvolvemos um experimento na *Web*, consistindo de seis páginas nas quais o usuário escolhe um artigo que seja familiar e responde um questionário, avaliando a qualidade das recomendações em diversas dimensões. As recomendações são geradas por um único algoritmo, que é aleatoriamente selecionado para cada usuário. Portanto, diferentes usuários podem receber recomendações de diferentes algoritmos, mas cada usuário recebe recomendações de somente um algoritmo.

Durante 32 dias, 110 usuários participaram do experimento. Eles foram convidados a partir de e-mails enviados a listas internas de universidades brasileiras e norte-americanas e através de links em sites que disponibilizam artigos científicos na *Web*. Para explorar diferenças entre idiomas, o sistema desenvolvido, chamado TechLens^{+}, possuía duas versões: uma em português e outra em inglês.Usuários eram solicitados a responder questões sobre cada recomendação e sobre o conjunto de recomendações. As questões buscavam explorar a satisfação dos usuários com as recomendações que foram geradas, a familiaridade com as recomendações e a descrição dos artigos recomendados. Considerando o conjunto total de recomendações, as questões buscavam também descobrir a satisfação dos usuários e qual o nível de experiência dos usuários.

Para avaliar a satisfação dos usuários, suas respostas foram categorizadas. Com relação à qualidade de cada recomendação e ao conjunto de recomendações, as respostas dos usuários foram classificadas indicando "satisfação" e "insatisfação" com relação às recomendações recebidas. Com relação à familiaridade com os artigos, os usuários foram considerados como "muito familiares", "familiares" e "não familiares". A tabela 2 mostra a satisfação dos usuários com relação a cada recomendação e ao conjunto de recomendações.

Tabela 2: Satisfação dos Usuários com as Recomendações

	Recomendações Individuais	**Conjunto de Recomendações**
Satisfeitos	46%	62%
Insatisfeitos	21%	19%

Com relação à familiaridade dos usuários com as recomendações, 27% delas eram muito familiares aos usuários, 34% eram familiares e 36% eram não familiares. Apenas 2% das recomendações foram escritas pelos usuários que estavam avaliando-as. Além disso, a satisfação das recomendações também variou por tipo de usuário, com 75% de satisfação de alunos de mestrado, 61% para estudantes de doutorado, 67% para pesquisadores e 52% para professores.

Analisando o tipo de artigo recomendado por cada algoritmo, Pure-CF e Fusion são melhores que o CF-CBF Separated para recomendar artigos novos[8] e autoritativos[9]. Para artigos introdutórios, CBF-Separated e CF-CBF Separated são melhores que Pure-CF. Por fim, CBF-Separated é melhor que Pure-CF que, por sua vez, é pior que todos outros algoritmos para recomendar artigos *survey*. A tabela 3 resume os melhores algoritmos para recomendar cada tipo de artigo.

Tabela 3: Algoritmos de Recomendação por Classe de Artigo

Classe de Artigos	**Melhores Algoritmos**	**Piores Algoritmos**
"Novos"	Pure-CF and Fusion	CBF-Separated
Autoritativos	Pure-CF and Fusion	CF-CBF Separated
Introdutórios	CBF-Separated and CF-CBF Separated	Pure-CF
Survey/Overview	CBF-Separated	Pure-CF

Aproximadamente 2/3 dos usuários que participaram do experimento *on-line* vieram ou do Brasil ou dos Estados Unidos. Um usuário é considerado desses países baseado de onde estava fisicamente localizado, obtido através de seu endereço IP. A satisfação individual das recomendações é praticamente a mesma, com 50% por parte dos norte-americanos e 49% por parte dos brasileiros. Insatisfação é parecida também, com 15% para norte-americanos e 17% para brasileiros. Por outro lado, a satisfação do grupo de recomendações variou bastante. Norte-americanos ficaram

[8] Novo, nesse caso, não necessariamente significa recente, mas algo desconhecido para o usuário

[9] Autoritativo significa que teve um grande impacto na comunidade científica. Geralmente, é um artigo bastante citado.

satisfeitos com 42% das recomendações e insatisfeitos com 33%, enquanto os brasileiros ficaram satisfeitos com 70% e insatisfeitos com apenas 12%.

Também houve variações em familiaridade. Norte-americanos eram mais familiares, com 31% muito familiares, 41% familiares e 24% não familiares. Brasileiros, por outro lado, foram 24% muito familiares, 31% familiares e 44% não familiares. Portanto, norte-americanos eram mais familiares com as recomendações e ficaram menos satisfeitos que os brasileiros.

Com relação à análise de idiomas, a versão em português do TechLens$^+$ iniciou seis dias após a versão em inglês. Brasileiros preferiram utilizar a versão em português após o seu lançamento. Dividindo então a população de brasileiros entre os que participaram das duas versões, observa-se uma grande variação. Com relação à qualidade do conjunto de recomendações, brasileiros no experimento em inglês ficaram satisfeitos em 42% e insatisfeitos com 33%, enquanto que no experimento em português, ficaram satisfeitos com 81% e insatisfeitos com apenas 3%. Por fim, no experimento em inglês, brasileiros foram 11% muito familiares, 32% familiares e 57% não familiares. Já no experimento em português, eles foram 29% muito familiares, 31% familiares e 40% não familiares.

3. Discussões e Conclusões

Nessa pesquisa nós descrevemos, implementamos e testamos diferentes técnicas para combinar Filtragem Baseada em Conteúdo e Filtragem Colaborativa para recomendar artigos científicos. Para testar os algoritmos, utilizamos experimentos *off-line* e *on-line*, em uma base de dados com 102.295 artigos.

Usuários que participaram em nosso experimento eram na maioria estudantes de mestrado e doutorado, pesquisadores e professores. Professores eram mais experientes que pesquisadores. Essa população provê respostas valiosas à nossa análise, que mostrou que quanto menos experiente for o usuário, mais ele gostou das recomendações. Além disso, professores eram mais familiares com os artigos, o que provavelmente os deixou "menos felizes". É importante salientar que 85% dos usuários consideraram no mínimo uma recomendação como sendo boa.

Considerando nossa análise cultural, nenhuma diferença significativa foi encontrada. Entretanto, brasileiros tiveram um percentual de estudantes de mestrado maior que norte-americanos. Brasileiros também ficaram mais satisfeitos, o que reforça a hipótese de que o nível de experiência influencia a percepção das recomendações (H3). Já na análise de idiomas, grandes diferenças foram encontradas. Brasileiros no experimento em português ficaram mais satisfeitos com as recomendações que os brasileiros no experimento em inglês.

Ainda com relação às hipóteses, FC e FBC podem ser combinadas com sucesso para recomendar artigos científicos (H1). Apesar de no experimento *on-line* as abordagens híbridas não terem tido a melhor satisfação, acreditamos que com o passar do tempo os algoritmos híbridos tenham um melhor desempenho, gerando recomendações mais balanceadas com as características das técnicas. Particularmente *Fusion*, que obteve o melhor desempenho *off-line.*Também foi verificado, de acordo com a tabela 3 que diferentes algoritmos são mais adequados para recomendar tipos diferentes de artigos (H2).

Os resultados obtidos nessa pesquisa nos levam a uma visão de um sistema de recomendação completamente personalizado sob três aspectos: em primeiro lugar, devem-se utilizar diferentes algoritmos para recomendar de acordo com o que o usuário busca, utilizando a tabela 3 como guia. Em segundo lugar, nossa análise de idiomas sugere que a interface deve ser na língua pátria do usuário. Finalmente, um sistema deve ser personalizado ao nível de experiência do usuário. Mais pesquisa é necessário para efetivamente determinar como realizar essa personalização, e como adaptar dinamicamente a interface para oferecer as recomendações de uma forma que não confunda usuários.

APPENDIX C TECHLENS+ SNAPSHOTS

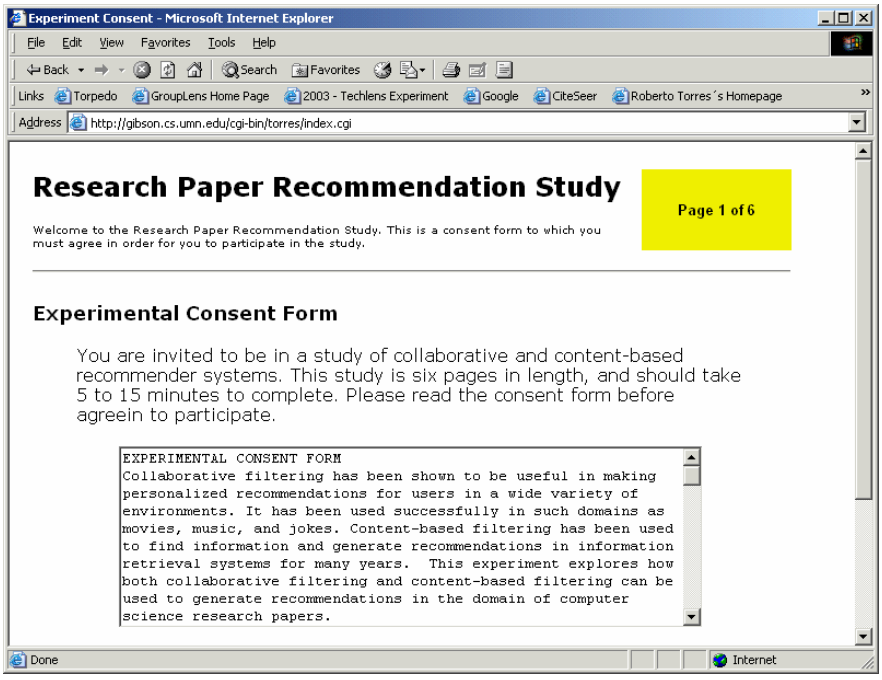

Figure C.1 – TechLens+ Consent Form

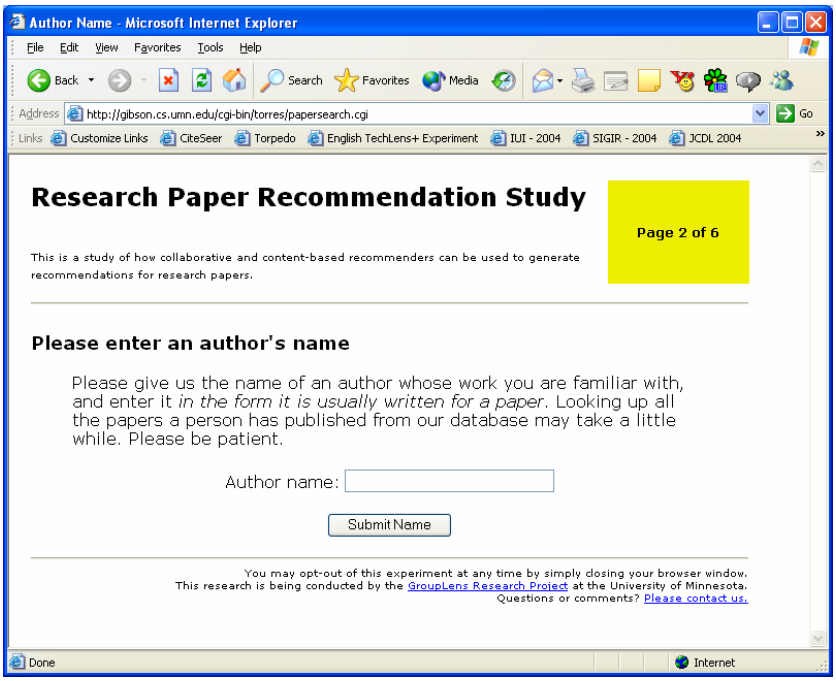

Figure C.2 – TechLens+ Author's Input

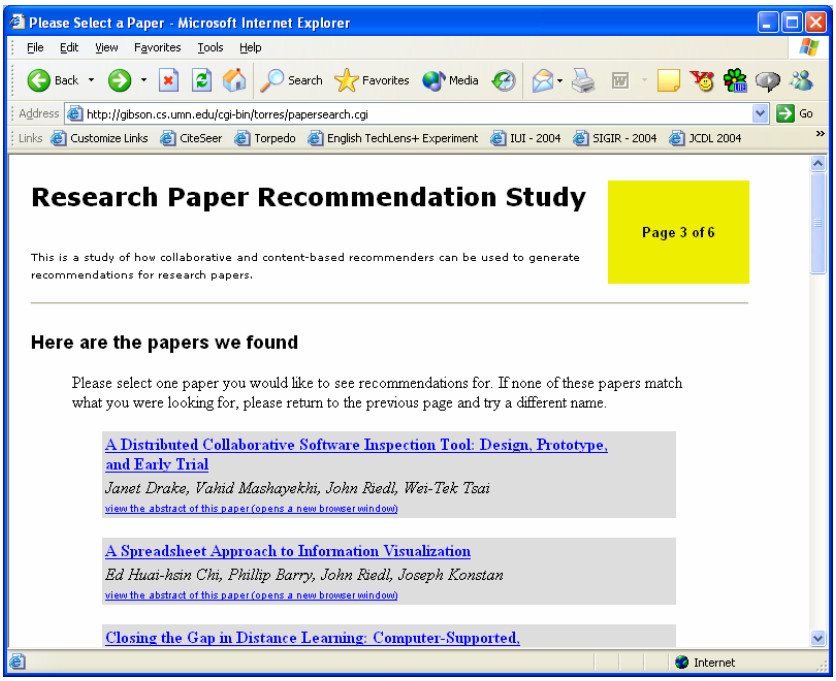

Figure C.3 – TechLens+ Paper Selection

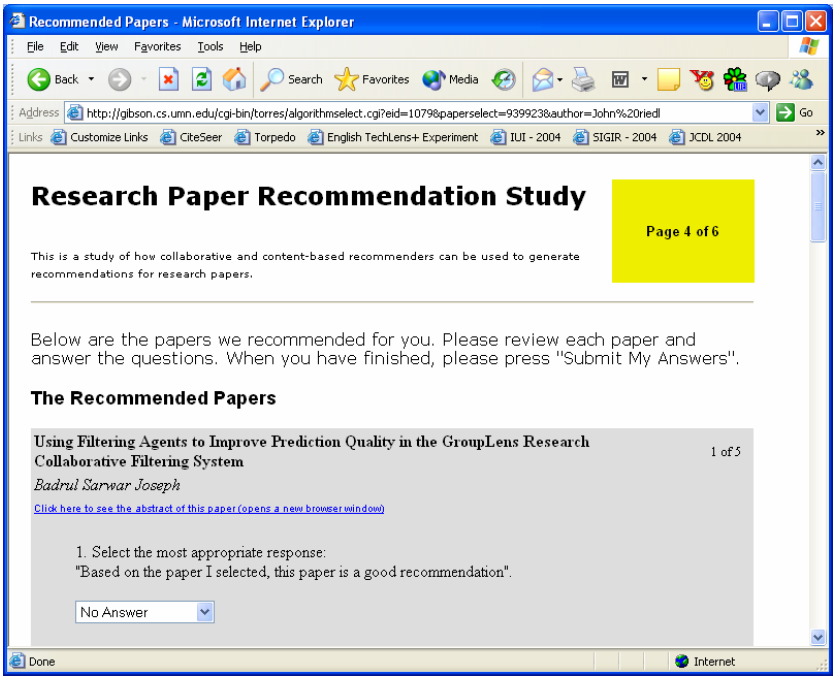

Figure C.4 – TechLens^{+} Recommendation's Evaluation

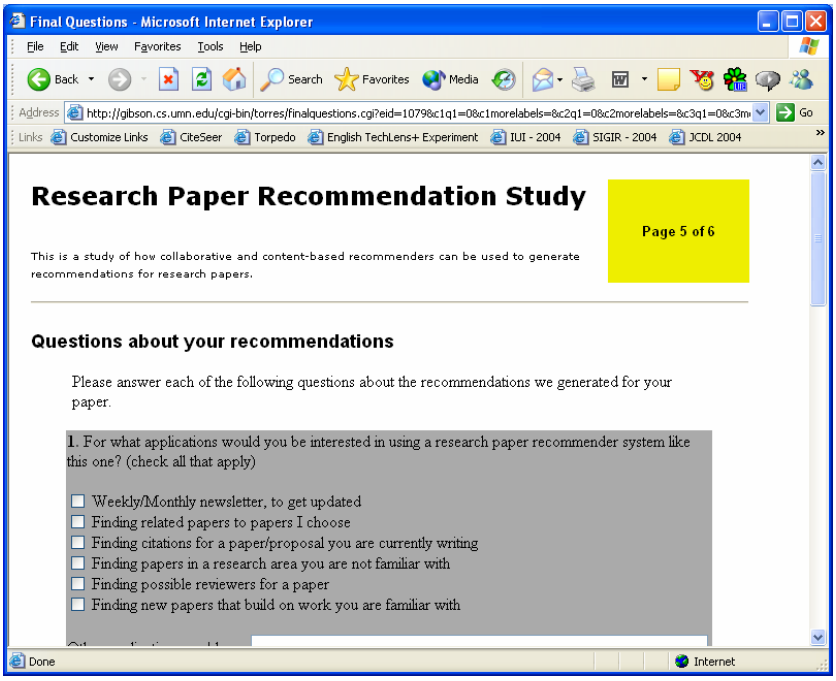

Figure C.5 – TechLens^{+} Overall Recommendation's Evaluation

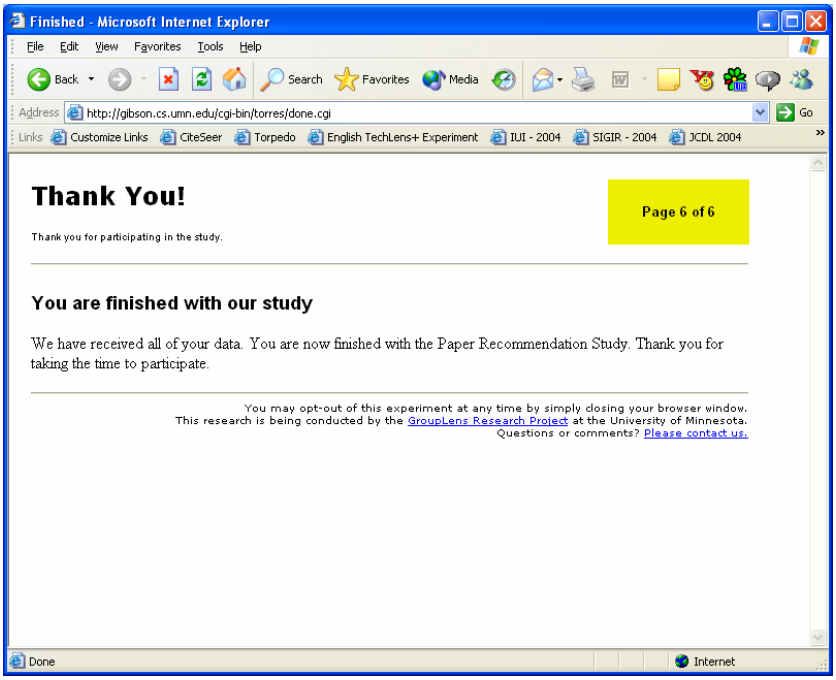

Figure C.6 – TechLens^{+} Ending