A Document Recommendation System Blending Retrieval and Categorization Technologies

Khalid Al-Kofahi, Peter Jackson, Mike Dahn*, Charles Elberti, William Keenan, John Duprey

{Thomson Corporation, R&D, *Thomson Legal & Regulatory, NPD}
610, Opperman Drive
St. Paul, MN 55123, USA

Peter.Jackson@Thomson.com, Khalid.Al-Kofahi@Thomson.com

ABSTRACT

The task of recommending documents to knowledge workers differs from the task of recommending products to consumers. Variations in search context can undermine the effectiveness of collaborative approaches, while many knowledge workers function in an environment in which the open sharing of information may be impossible or undesirable. There is also the 'cold start' problem of how to bootstrap a recommendation system in the absence of any usage statistics. We describe a system called ResultsPlus, which uses a blend of information retrieval and machine learning technologies to recommend secondary materials to attorneys engaged in primary law research. Rankings of recommended material are subsequently enhanced by incorporating historical user behavior and document usage data.

Introduction

The task of recommending documents to knowledge workers differs from the task of recommending products to consumers. Collaborative approaches [1, 2, 3], as applied to books, videos and the like, attempt to communicate patterns of shared taste or interest among the buying habits of individual shoppers to augment conventional search results. There are well-known problems with these approaches, e.g., when consumers temporarily shop for their children, but their effectiveness has been established in practice at ecommerce sites such as Amazon.

It turns out that subtle variations in search context can undermine the effectiveness of collaborative filtering. For example, a lawyer might research one side of a case today, and tomorrow want to argue the other side of a similar case. This is rather like the 'shopping for children' example, in which a consumer's tastes and interests appear to change capriciously, from the system's point of view. Also, lawyers are reluctant to share their search history with others for a variety of reasons, ranging from confidentiality to competitive advantage.

However, there are also obvious commonalities between these different types of 'shopper', in that recommended items, whether documents, clothing, or videos, must add value to the result list derived from conventional search. A certain level of accuracy or appropriateness is also required in order to gain the consumer's trust. In what follows, we assume that recommendations are based solely upon their content or properties, and not any advertising mechanism involving paid inclusion in a search result, or the promotion of new, sale, or discount items.

Consumers of information typically rely upon classification schemes as an adjunct to search, browsing through taxonomies and tables of contents to narrow the application of queries. The problems with such approaches are also well known, e.g., the inflexibility of taxonomical organizations of knowledge and the fact that documents can belong to multiple categories to different degrees. Nevertheless, it makes sense to consider the role of classification in the construction of recommender systems, and the role of linear classifiers in particular, given their effectiveness for text categorization [4, 5]. Some studies show that linear classifiers can outperform memory based collaborative filtering approaches to recommendation tasks and have better computational properties [6, 7].

For information seeking, what seems to be required is a document recommendation system that takes into account both the user's particular query and certain features from the overall search context. One set of such features might include any metadata, such as subject matter classifications and citation patterns, associated with the documents themselves. Being able to leverage existing taxonomies and inter-document relationships helps address the 'cold start' problem, where no user data exists initially, especially where the number of items (documents) greatly exceeds the number of users. Another set of features includes the click-through behavior of both individual users (for personalization) and groups of users (to smooth sparse usage data).

We describe a system called ResultsPlus, which uses a blend of information retrieval and machine learning technologies to recommend briefs and secondary law materials to attorneys engaged in primary law research. The system incorporates historical user and document usage data to further enhance the ranking of its recommendations. Briefs are documents written by attorneys to present their legal arguments in a court proceeding. Secondary materials include articles from legal encyclopedia, legal research papers, and law reviews. ResultsPlus has been successfully implemented in production as a document recommendation feature on Westlaw, and is now applied to all case law searches.

The underlying technology employed is a text categorization framework called CaRE [11], which combines multiple classification algorithms to enable good performance on large numbers of categories (> 100,000). We first show how a highly scalable text categorization system can use existing taxonomies to make accurate recommendations, as measured by suggestion rate, precision, and a few other measures specific to the application. We then show how the addition of user data can demonstrably improve how recommendation candidates are ranked.

The Legal Domain

Any common law system relies heavily upon the written pronouncements of judges to interpret the law. Each judicial opinion not only attempts to resolve a particular legal dispute, but also to help resolve similar disputes in the future. Therefore, judges and lawyers are continually researching an ever-expanding body of case law for past opinions that are relevant to the resolution of a new dispute.

To facilitate these searches, some legal publishers not only collect and publish the judicial opinions of courts across the United States, but also summarize and classify the opinions based on the principles or points of law they contain. For example, Thomson creates *headnotes* for each case, which are short summaries of the points made in judicial opinions. A typical judicial opinion is allocated about 7 headnotes, but cases with hundreds of headnotes are not rare. On average, about 500,000 new headnotes are created each year, and our repository contains over 22 million.

Headnotes are classified to the West Key Number™ System; a hierarchical classification of the headnotes across some 100,000 distinct legal categories, or classes. Each class has not only a descriptive name, but also a unique alpha-numeric code, known as its *Key Number*.

In addition to using highly-detailed classification systems associated with *primary* law (cases and statutes), judges and lawyers also conduct research using *secondary* materials, such as American Law Reports (ALR), that provide in-depth scholarly analysis of a broad spectrum of legal issues. ALR includes about 14,000 distinct articles, known as *annotations*, each addressing a separate legal issue, such as double jeopardy or free speech. Each annotation also includes citations and headnotes identifying relevant judicial opinions to facilitate further legal research.

Another example is American Jurisprudence (AMJUR), a legal encyclopedia with A-to-Z type coverage of the law. AMJUR is organized into about 130 topics (e.g., Family Law, Criminal Law), with each topic organized into chapters, sections and sub-sections. Overall, AMJUR contains about 135,000 sections (or categories).

To ensure currentness, analytical law products, such as ALR and AMJUR, are continually updated, so that they cite new recent judicial opinions as they are published. To be more precise, analytical law articles only cite those issues in a case that are relevant to them; as typically a case has several issues not all of them are relevant to an analytical article. For historical reasons, this process of citation enrichment is called *supplementation*. Because headnotes represent the issues in a case, supplementation is, for the most part, a headnote classification problem. In addition to creating references from secondary law products to cases, cross-references are also created between the secondary law products themselves. Thomson publishes several hundred such products.

Thus, an information system that can suggest good secondary sources to a user searching case law is providing a useful service. An on-point secondary source will summarize a particular area of law, cite all important cases, and provide links to related secondary materials, e.g., specialty publications on particular topics, such as bankruptcy or tax law.

Traditionally, all the classification and linking tasks described above have been done manually, but over the last six years such tasks have increasingly been assisted by our CaRE text classification system. Since this is also the system behind our document recommendation engine, we describe it in some detail in the next section.

The CaRE System

Research has demonstrated that superior automatic classification can be achieved through a combination of multiple classifiers. For example, Larkey & Croft [8] have shown that the weighted sum of three classifiers is superior to a combination of two classifiers, which is superior to each of the individual classifiers. The classifier weights were proportional to the performance of the individual classifiers. Iyer et al. [9] employed boosting to combine many weak classifiers for text filtering. The resulting classifier compared favorably with a modified version of the Rocchio algorithm. Tumer and Ghosh [10] used order

statistics to combine several classifiers resulting in superior classification. Both voting and averaging methods have been shown to improve performance. The rationale is that averaging reduces the classification (i.e., score) variance, which decreases the overlap between the scores of relevant and non-relevant documents.

CaRE (Classification and Recommendation Engine) is a generalization of CARP [11], a program that classifies newly written case summaries to sections of American Law Reports for citation purposes. The framework has all necessary functionality for extracting features from documents, indexing category profiles, storing the profiles into databases, and retrieving them at run time for classification. It comes equipped with a pool of existing feature extractors, classifiers, meta-classifiers, and decision makers.

The *feature extractors* can handle words, word-pairs (not typically bigrams but rather word pairs within a text window of a certain size), and various meta-features such as citations and key numbers.

The *classifiers* consist of Vector Space, Bayesian, and KNN modules. These approaches have been widely reported in the literature, so we make no effort to analyze them here. Users can configure the system at run time by selecting which classifier to use on which feature type.

Meta-Classifiers operate on the output of the classifiers themselves and combine their scores. Different meta-classifiers are available, such as simple averaging, weighted averaging, and assigning classifiers different weights on a per category basis.

The *decision maker* is responsible for the actual classification. Rules may be incorporated into this module, e.g., take the *n* best-scoring suggestions from the meta-classification stage.

Thus, for the supplementation of American Law Reports, all ALR articles were first indexed with respect to the words and word pairs they contained as well as key numbers occurring in their extant citations.

To classify new headnotes to these articles, CaRE uses two classifiers combined with two different feature sets to yield four combinations. The classifiers are Vector Space and Naïve Bayes, and the feature sets are headnote text and their associated key numbers. One can consider the combinations as four different similarity measures between headnotes to be routed and candidate ALR articles they could supplement.

A headnote is represented by a set of all non-stopword pairs present in it. Since a headnote is short and focused on a well-defined point of law or factual situation, some of the word pairs can be thought of as approximating key concepts. E.g., the co-occurrence of 'drug' and 'school' can be taken to represent the idea of drugs being taken into schools, or sold near schools.

The second feature set consists of 'leaf' key numbers only. Intermediate key numbers, along with their implied vertical and horizontal relationships in the hierarchy are ignored. In-house statistical studies have shown that proximity in the key number hierarchy does not necessarily imply closeness among the corresponding concepts

Given the four similarity measures assigned to a headnote-annotation pair, S_i, for i = 1 to 4, the similarity between headnote h and annotation a is estimated by

$$S_a^h = \sum_{i=1}^{4} w_{ia} S_i ,$$

where, w_{ia}, is the weight assigned to classifier i and annotation a. Headnotes are then assigned to annotations according to the following decision rule:

$$\textit{Assign headnote, h, to annotation, a, iff } S_a^h > \Gamma_a$$

where Γ_a is an annotation-specific threshold that was determined based on a held-out tuning set.

Further details can be found in [11]. The use of CaRE in ResultsPlus is rather similar; differences will be noted in the next section.

Generating Recommendations

ResultsPlus recommendations are generated using a two step process: *generation* followed by *optimization*. In the first step, a ranked list of recommendations is generated using content-based similarity (CaRE). In the second step, recommendations are re-ranked based on user behavior and document usage data.

The Generation Module (GM) treats queries as if they were snippets of text in need of classification to secondary law articles and briefs. Thus, we are using CaRE as a search engine. Queries are run in this way against multiple indexes, each representing a different set of articles, such as American Law Reports, American Jurisprudence, various Law Reviews, and so on.

The main difference between how GM works and the supplementation of ALR is that we no longer use the key number feature set. Instead, we 'enrich' the text of each candidate article with the text of the most relevant case summaries that it references. Each set of articles is then treated as a separate database to be queried against, and is indexed by both words and word pairs, as before.

At query time, queries are 'featurized' into words and word pairs, and run against all databases by applying both Vector Space and Naïve Bayes classifiers to the resulting features. The scores

returned from these document sets, while consistent within each set, they are not directly comparable across the board.

Consequently, an offline normalization step is needed. The normalization functions were computed offline by compiling the recommendations from each product for a large set of user queries, computing the accumulative histograms of the scores, and then fitting gamma functions to the resulting histograms. Each gamma function is then used to translate a publication-specific score into one that uses the same yard-stick across all publications and articles.

The recommendations derived from each classifier are then aggregated into one set (by multiplying their normalized scores), ranked according to their normalized scores, and a global threshold is applied to ensure high quality. Typically, less than 20 recommendations exceed the threshold.

Recommendation Optimization

The problem of optimizing the performance of a ranking algorithm has received significant attention in the literature, and seems analogous to maximizing retrieval or classification performance. In this sense, a likely metric of ranking performance would be average precision of documents retrieved at particular ranks. Yet average precision is essentially a composite of binary judgments and does not capture how one relevant item might compare with another in terms of importance [13][14].

More appropriate metrics compare rankings produce by a given function with an ideal ranking. Joachims [14] utilizes Kendall's τ to measure how closely an ordering approximates to the ideal. Other research incorporates similar ideas in building criteria function [15][16][17], in that evidence is accrued by comparing orderings of all possible pairs in a ranking, the assumption being one item should always be preferred over another.

Rank optimization algorithms tend to fall into two broad categories. Some algorithms measure performance against a subset of the original training collection to optimize scoring or ranking functions incorporating search methods such as genetic programming or gradient-based optimization to maximize ranking performance [13][18][19][20]. Other algorithms employ user feedback in the form of query logs (click through data) to discover and leverage underlying patterns in customer behavior [21][22].

Algorithms that utilize the training corpus for ranking optimization primarily seek to improve overall performance by combining evidence from multiple distinct language models, classifiers, and weighting schemes [13][20][18]. For example, Fan et al. [18] describe a ranking function discovery framework that uses genetic algorithms to search for optimal term weighting schemes.

Algorithms that utilize click through data to construct optimal ranking functions focus on mining query logs to solve the ranking problem. This data is particularly amenable to "relative relevance" metrics such as Kendall's τ mentioned earlier. Common themes in this research included customer-aware searching, query disambiguation, data fusion of meta-search results, and co-training/clustering methods for sparse data collections [21][22].

Our approach for rank optimization relies on historical user and document data.

It estimates the expected click through rate (CTR) for each recommendation and ranks the list accordingly. The ranking algorithm relies on six context specific features: (1) the user ID, (2) the suggested document/article ID, (3) the publication type of the suggested document, (4) the jurisdictions the user has selected, (6) the databases the user is searching, (7) the user's location, and the user's market segment.

In an *offline* process, the historic CTR data is collected and aggregated for several combinations of these features, including (1) document click through rate, (2) user-specific click through rate for a given publication type in general and relative to the queried database and jurisdiction, (3) group-level CTR for a given publication type in general (4) all-user CTR for a given publication type in general and relative to the queried database and jurisdiction. These combinations enable us to use very specific CTR data for a given document or a user's current search context. When there is insufficient history (support) for a specific CTR combination, a more general (group or all-user) based CTR is employed.

The basic formula for calculating the historic CTR is number of clicks per suggestion. Clearly, such a formula is biased because users tend to click on top ranked documents more than those at lower ranks. Therefore it is necessary to normalize the number of clicks by the rank of the documents. This is achieved by dividing click counts by the CTR of that rank. For example, if rank 1 gets twice the number of clicks as rank 2, a click at a rank 2 suggestion is assigned twice the 'value' of a rank 1 click, and so on.

The *online* recommendation engine uses the current user's profile attributes and search context (the features described above) and retrieves the previously calculated CTR values from a database. Thus, several historical CTR values are considered by the ranking algorithms. In general, the most contextually specific CTR value with sufficient support is selected. For example, if we have a CTR value that indicates a user finds publication type A useful, that value is selected over the baseline CTR for all users. Essentially, we try to accumulate statistics about every possible useful combination of user-data clicks, and we use back-off

procedures to account for missing historical data or for new publications.

In order to avoid any inherent bias against new documents in our approach, and to avoid a scenario where a class of documents is not suggested at all, we randomly will boost a small number of documents from ranks 11-20 to ranks 3-10 for a small percentage of the queries, along the lines suggested in [23].

System Performance

In the information retrieval community, search methods are usually evaluated using recall and precision measures, or functions of these measures (e.g., the F1 measure). Search methods are also evaluated based on their ability to rank relevant documents before non-relevant ones. Some of the evaluation methods rely on binary relevance judgments, while others permit varying degrees of relevance.

In this work, precision is more important than recall, because a recommender system is not designed to suggest all relevant items, but only some of them. In fact, in order to avoid overwhelming users with recommendations, we impose an upper limit on the number of recommendations made. Suggestion rate, or the percentage of queries that elicit recommendations, is more important than recall, because it is more correlated with the system's utility. Overall, we identified a set of nine statistics (see Table 1) that were deemed to be relevant to system's performance, which we shall discuss below.

In addition to these nine metrics, we measure the effectiveness of the rank optimization algorithms by their impact on click through rate.

Experimental Design

In designing the evaluation experiment, we made the following decisions.

First, we decided to use a 3-point relevance scale of *on-point*, *relevant*, and *not relevant*, denoted as 'A', 'C', and 'F', respectively. The assumption was that both testers and end users would be able to distinguish reliably between As and Cs; we will need to revisit this assumption below.

Second, we only used 'well-written' queries in our evaluations. Well-written queries are defined as those that are both indicative of a topic and have limited scope. Many Westlaw queries are not well written by any standards, and the business would have liked some measure of the system's performance on these queries. But in practice it is extremely difficult (if not impossible) to provide relevance judgments for vague queries. In the end, the system was designed to recognize vague queries and not make any recommendations for them. We used a set of simple heuristics for this purpose (e.g., the number of features extracted from a query, their Inverse Document Frequency).

Third, to reduce inter-assessor variability, each recommended article was judged by three assessors independently of each other. In total, five assessors were used. The assessors are all 'reference attorneys' who assist online customers searching Westlaw content by helping them craft queries. In the experiment reported below, we used the median score of the three assessors as the 'gold standard' judgment.

In order to promote consistency between the assessors, we held a discussion group before the actual exercise and we tried to establish guidelines for the different levels of relevance. Still, inter-assessor agreement was generally low (in the 50% to 60% range), especially on the fine distinction between the 'relevant' and 'on-point' categories. This is perhaps unsurprising in the light of earlier studies [12].

To evaluate the system, we selected a set of 650 well written queries from the query logs The assessors were not involved in the selection process to avoid any bias. We tried to select queries that represented disparate topics. However, we did not consult the target content to determine these topics.

Table 1. Nine Statistics Used to Assess ResultsPlus

Measure	Value
Queries with at least one grade *A*	87.90%
Queries with mostly *A*s and *C*s	84.40%
Queries with at least one *A* or one *C*	**96.01%**
Queries with no *F*s	72.35%
Queries with all *F*s	**3.98%**
Total *A*s	66.59%
Total *C*s	19.78%
Total *F*s	13.63%
Queries with suggestions	**88.54%**

Table 1 lists the values for the nine metrics. Bold type distinguishes those metrics deemed most important by the business.

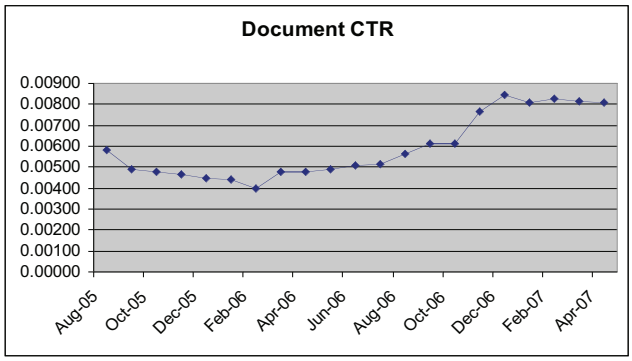
A couple of observations are in order. First, one should avoid generalizing the above numbers to queries of mixed

quality, at least without any qualifications. Second, the high inter-rater disagreement, even on good queries, discouraged us from using queries of mixed quality for estimating the system's performance. Such a test would have suffered from even higher inter-rater variability, and the resulting statistics would have been deemed unreliable.

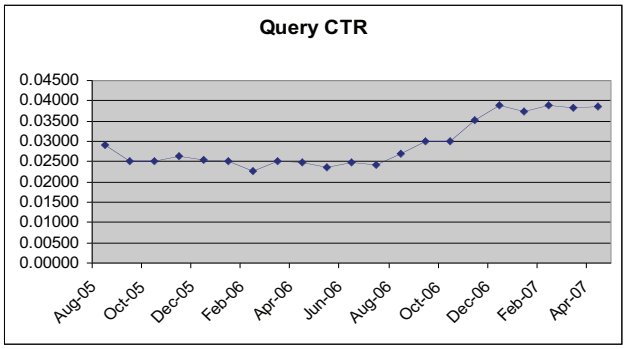
We measure the effectiveness of the rank optimization algorithm by its impact on click through rate. In particular, we use three click-through measures. Document CTR, query CTR (defined as the average number of clicks per query, or result set), and session CTR (defined as the average number of clicks per user session). These values of these metrics are plotted in Figures 1 (a), (b), and (c).

In reviewing these figures, keep in mind that the first rank optimization algorithm was introduced in September of 2005, and during this period we tried 29 different versions of the ranking algorithms. We made significant improvement in August of 2006, and we incorporated our current rank optimization algorithm into production in November of 2006. These are the dates to look for in the figures.

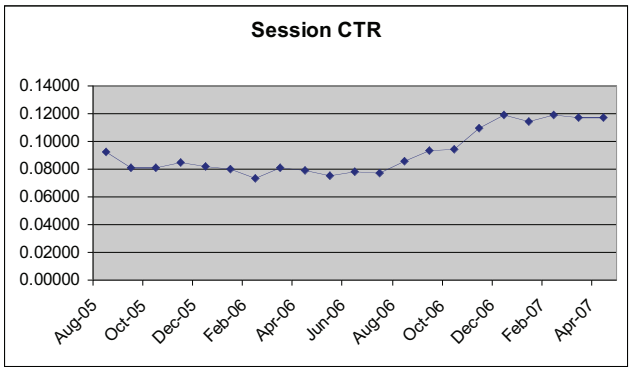
Beyond the visible increases in CTR around August and November of 2006, we are unable to explain the monthly variations in CTR. The graphs highlight the fact that it took us about one year to gather enough statistics and tune the ranking algorithms to achieve higher click through rates. The fact that ResultsPlus was part of a much larger system that introduced its own variables into the mix complicated things. Another complicating factor is the fact that we increased the size of the recommendations pool by several factors during the same time period. The lack of document usage data forced the system to back-off to publication-level and database-level usage statistics, which too were very sparse for these newer publications.



(a)



(b)



(c)

Figure 1: Click through rates for (a) documents, (b) queries, and (c) sessions, for the period August 2005 through April 2007

User Studies

We held 6 offsite focus group studies involving a total of 44 attorneys. 29 of those attorneys identified themselves as litigators, while the remaining 14 identified themselves as transactional attorneys. The purpose of these studies was to measure user expectations, gauge their reaction to this new system, and to estimate the click-through rate as a function of their relevance judgments. We describe this study here, because we believe it provides important insights over and above the results described in the previous section.

We started each focus group by explaining to the participants that a query could cover multiple subjects, and asked about the general characteristics of the documents that they thought should be returned. We then used a specific example, and asked about the documents that should be returned for that example. Afterward, we had participants construct a sentence describing their expectations.

Next, we presented users with 18 sets of ALR articles, each with a corresponding legal issue and a query, and asked them which articles were relevant to the research and which they would click on. We also asked whether we should

have shown all of the ALRs, some of the ALRs, or none of the ALRs.

We performed several experiments of this type, where we asked our focus group to *describe* their expectations, and then to *show* us their expectations through examples. Interestingly, users had much higher expectations when they *told* us what they expected as compared to when they *showed* us what they expected.

We also performed several experiments where the result set (3 suggestions only) consisted of a mixture of on-point (A), relevant (C), and not-relevant (F) results (e.g., 3A, 2A+1F, 1A+2C). We then presented the result sets with the corresponding research issue and query to the participants. The click-through rates on the 'A' and 'C' recommendations were almost identical.

Summary and Conclusions

Our experiments (and a successful product launch) show that state of the art document categorization algorithms embedded in a scalable architecture can deliver high performance in terms of both accuracy and throughput for a commercial recommender system. Westlaw users have embraced ResultsPlus, generating significant additional revenues, since not all documents that are suggested will be in a user's subscription plan. Although some users will never go 'out of plan', others are willing to purchase highly relevant documents on an occasional basis.

We believe that ResultsPlus demonstrates the effectiveness of a multi-classifier approach in overcoming the 'cold start' problem for a very sparse user-document matrix. In a relatively closed user community, such as Westlaw, it is important that recommendations are perceived as relevant from the very beginning, else widespread adoption may never take place. In our case, much of the product metadata used to launch the system was itself generated by a machine learning system.

In a critique of [14], Cao et al. [24] point out that the difference vector approach to ranking that we have employed does not distinguish between high-rank and low-rank inversions. Ideally, one would like to penalize inversions in the higher ranks more than inversions lower down in the ranking. One approach attempts to optimize Normalized Discounted Cumulative Gain at a given rank by heuristically setting costs.

Even so, the ResultsPlus experience has taught us the value of user data and its potential for improving search rankings based on keywords and document metadata alone. Properly implemented, learned ranking functions can be derived and deployed efficiently for moderately sized feature sets using Support Vector Machines. Storage costs are quite acceptable and recommendations can be generated at least as quickly as normal search results, resulting in no performance penalty.

We do not believe that our findings are restricted to the present domain of legal information. Although some of the content we recommend is both high quality and highly editorialized, other content, such as legal briefs, are more variable and relatively ungroomed. Our experience on Westlaw can be seen as validating the smaller scale experiments on search engine optimization that have been reported in the literature.

References

[1] Twidale, M. B., Nichols, D. M. & Paice, C. D. (1997). Browsing is a collaborative process. *Information Processing and Management*, 33(6), pp. 761-783.

[2] Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., Riedl, J. (2002). Getting to Know You: Learning New User Preferences in Recommender Systems. In *Proceedings of the 2002 International Conference on Intelligent User Interfaces (IUI-02)*, New York: ACM Press, pp. 127-134

[3] Boutilier, C., Zemel, R. S., and Marlin, B. (2003) Active Collaborative Filtering. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI-2003)*, pp. 98-106.

[4] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML-98)*, pp. 137-142.

[5] Yang, Y. & Chute, C. (1994). An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12, pp. 252-277.

[6] Breese, J., Heckerman, D. & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-1998)*, pp. 43-52.

[7] Zhang, T. & Iyengar, V. S. (2002). Recommendation systems using linear classifiers. *Journal of Machine learning Research*, 2, pp. 313-334.

[8] Larkey, L. and Croft, W. B. (1996) Combining Classifiers in Text Categorization. In *Proceedings of The 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-1996)*, pp. 289-297.

[9] Iyer, R. D., Lewis, D. D., Schapire, R. E., Singer, Y. & Singhal, A. (2000). Boosting for document routing. In Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM-2000), pp. 70-77.

[10] Tumer, K. & Ghosh, J. (1999). Linear and order statistics combiners for pattern classification. In Sharkey, A. (ed.) *Combining Artificial Neural Networks*, Springer Verlag, pp. 127-162.

[11] Al-Kofahi, K., Tyrrell, A., Vachher, A., Travers, T. & Jackson, P. (2001). Combining multiple classifiers for text categorization. In Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM-2001), pp. 97-104.

[12] Voorhees, E. M. (2001). Evaluation by highly relevant documents. In *Proceedings of The 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2001)*, pp. 74-82.

[13] Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. Automatic Combination of Multiple Ranked Retrieval Systems (*SIGIR 1994*), pp. 173-181

[14] Thorsten Joachims. Optimizing Search Engines using Clickthrough Data (*KDD 2002*), pp. 133-142

[15] William W. Cohen, Robert E. Shapire, and Yoram Singer. Learning to Order Things. J. Artif. Intell. Res. (JAIR) 10: 243-270 (1999)

[16] Thorsten Joachims. Evaluating Retrieval Performance using Clickthrough Data (*Text Mining 2003*), pp. 79-96

[17] Qingzhao Tan, Xiaoyong Chai, Wilfred Ng, and Dik-Lun Lee. Applying Co-training to Clickthrough Data for Search Engine Adaption (*DASFAA 2004*), pp. 519-532

[18] Weiguo Fan, Michael D. Gordon, and Praveen Pathak. A Generic ranking function discovery framework by genetic programming for information retrieval. Inf. Process. Manage. 40(4): 587-602 (2004)

[19] Joon Ho Lee. Combining Multiple Evidence from Different Properties of Weighting Schemes (*SIGIR 1995*), pp. 180-188

[20] Paul Ogilvie, Jamie Callan. Combining Document Representations for Known-Item Search (*SIGIR 2003*), pp. 143-150

[21] Rodrigo B. Almeida, Virgilio A. F. Almeida. A Community-Aware Search Engine. (*WWW 2004*), pp. 413-421

[22] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan. Optimizing Web Search Using Web Click-through Data (*CIKM 2004*), pp. 118-126

[23] Pandey, S., Roy, S., Olston, C., Cho, J. & Chakrabarti, S. (2005). Shuffling a stacked deck: The case for partially randomized ranking of search engine results. *Technical report CMU-CS-05-116*, School of Computer Science, Carnegie Mellon University.

[24] Cao, Y., Xu, J., Liu, T.-Y., Li, H., Huang, Y. & Hon, H.-W. (2006). Adapting ranking SVM to document retrieval. *SIGIR-2006*, pp. 186-193.