Finding Information:

Intelligent Retrieval & Categorization

Dr. Jan Puzicha, Chief Technology Officer, Recommind Inc.

For more information contact:
+1-415-394-7899 or info@recommind.com
Or visit Recommind at http://www.recommind.com
1. Drowning in Information Chaos
2. Managing Information: Establishing Context
   2.1 Determining Information Activities
   2.2 Identifying Key Success Factors
   2.3 Developing the “Best-Fit” Solution
3. Information Systems: Retrieval and Categorization
4. Information Retrieval Technologies
   4.1 Keyword-Based Search
   4.2 Vector-Space Retrieval Models
   4.3 Concept-Based Retrieval
   4.4 PLSA (Probabilistic Latent Semantic Analysis)
   4.5 Advantages of CORE’s PLSA-based approach
5. Categorization Technologies
   5.1 Manual Categorization
   5.2 Rule-Based Categorization
   5.3 Naïve Bayes Categorization
   5.4 Nearest Neighbor Classification
   5.5 Support Vector Machine Categorization
   5.6 PLSA
6. Personalization and Collaborative Filtering
7. Performance Comparison: Algorithms
   7.1 Retrieval Performance
8. Conclusion
9. Bibliography
1 Drowning in Information Chaos

Recent advances in computing, networking, and data storage capabilities have made a huge amount of information accessible to us all. The data we have at our finger tips today may well contain the name of the individual and expertise we're trying to find, the molecule we're trying to identify, or the story we're trying to communicate. But having that information available is not enough--we need to find it, master it, and make it useful. In the words of a study performed by researchers at UC Berkeley, “We are all drowning in a sea of information--the challenge is to learn how to swim.”1

While the challenge of the task is very great, the benefits of swimming in information (as opposed to the drowning most of us are now doing) are also great. Early companies to adopt infrastructures for dealing with knowledge latency in their organizations have reaped very large benefits in terms of direct cost savings and substantial productivity improvements. A clear example of an early adopter realizing substantial cost saving from the deployment of knowledge management infrastructure is BP Amoco (BP). BP’s “Shared Learning” knowledge management program has saved the company nearly $700 Million in its first two years. On one North Sea drilling project alone, team leaders saved $80 Million by applying cost-saving tips they learned from experts around the company. Similar advances have been observed in the collaboration and productivity with which employees perform their work tasks--at Boston Consulting Group, for example, “It now takes...three or four hours to put something together that used to take three or four days.”

Viewed in its broadest context, the challenge facing organizations like BP and BCG can be articulated as simply that of helping employees find the right information to make decisions or perform tasks. In many critical cases, finding specific information is the task. Nevertheless, employees access and utilize information differently. For example, publishing organizations may discover and push relevant news releases and flashes to the appropriate editorial desks,

---

while operation managers at oil companies may actively search across the organization for new tools for analyzing geologic data. These use cases reflect differences in task workflow and the way in which information is accessed or made available, but the core task is essentially the same: finding relevant information.

2 Managing Information: Establishing Context

- What then should we look for in determining our information management needs? We can begin by answering several key questions: What task or tasks are being undertaken by the users of the system?
- What will the key success factors be in the deployment of the system?
- What tools should be deployed and what do they need to look like in order to perform these tasks being undertaken effectively?

2.1 Determining Information Activities

Information activities can be divided into two broad categories: (i) user-initiated retrieval in response to a query and; (ii) pre-emptive categorization and labeling before a user requests the information. In many cases, sufficiently powerful retrieval capabilities render prior categorization unnecessary. In others, (system initiated labeling, for example) no user query is made to guide retrieval, so documents must be categorized for access later. In both cases the amount of information available today necessitates the organization (categorization) of results into topics relevant to the user.

2.2 Identifying Key Success Factors

Having determined the process workflow, and thereby the broad tools required, what key factors will drive the success of the information system? Here, the two most important factors are system accuracy and automation level. Accuracy determines the effectiveness of task performance versus automation determines the usability of the system.
The costs of poor accuracy cannot be underestimated. In an IDC study of corporate executives, 39% of participants spent more than nine hours a week searching for information, and 69% spent five or more hours per week. Of these, 68% responded that they succeeded in finding what they were looking for less than 85% of the time--at least five hours of lost time each as a result of poor accuracy. Information system accuracy is essentially equivalent to productivity.

Information systems have their own costs, however, related initially to the indexing of information and system training, and later to the ongoing maintenance required for a dynamic (and in almost all cases growing) corpus. These costs are composed of direct costs in terms of system resources (hardware and software) but also in terms of what can be very substantial indirect costs. These indirect costs are formed primarily by the time invested by users in the maintenance of taxonomies, the labeling of documents, and in some cases manual training of the system.

2.3 Developing the “Best-Fit” Solution

Given the success factors outlined above, we can draw some important conclusions. First, the tools deployed must be appropriate for the information task and adaptable to the appropriate domain and workflow. Second, accuracy in performing the information task must be maximized in the deployment of a system, as this is the key driver of user productivity in the performance of the task. Finally, ease of system use and deployment must be maximized to increase user adoption and to minimize productivity drains associated with deploying the system. Ease of system use is driven by the automation of indexing and training. Ease of deployment is also driven by the automation of indexing and training, as well as the ease of integration with other systems within the workflow. The best product for any given task will provide the most effective balance of automation and accuracy for the task at hand, provided it integrates well within the system architecture.
3 Information Systems: Retrieval and Categorization

The framework above provides a context for thinking about information retrieval and categorization systems. The effectiveness of these systems will be determined by 1) the work task; 2) the accuracy of the system; 3) the degree of automation in indexing, training, and ongoing maintenance that can be achieved; and 4) the ease of integration with existing systems.

There are many approaches to solving these problems, ranging from the manual creation of taxonomies and the manual categorization of documents into them, to much more sophisticated automated approaches that have been developed recently. Each of these technologies has strengths and weaknesses with respect to the accuracy and the automation that can be achieved, but in some cases the trade-offs are less significant. A brief overview of common commercial technologies used for information retrieval and categorization follows.

4 Information Retrieval Technologies

More efficient search by intelligence and interaction

Information retrieval based on query terms is probably the most popular form of information access. It is used in cases where the user has an immediate information need and this need can at least be expressed to a first approximation in a simple text query.

4.1 Keyword-Based Search

In keyword-based search, the query terms are matched against the document collection by identifying those documents that contain the requested terms. Matching documents form the result set. Documents in the result set are then ranked according to ranking functions that typically depend on the frequency of the query terms in documents or on external information like the number of incoming links in the context of Web search.
The focus in the design of keyword-based search engines has been on scalability and data dynamics. How can systems achieve real-time responses, even for collections that may exceed several million documents? For that purpose, efficient indexing data structures have been developed that avoid searching through documents by an off-line process that generates and maintains a suitable index. In the simplest case, this will be nothing more than a table that contains for each term the corresponding documents in which that term occurs.

The limitations of keyword-based search are obvious. Since each term is treated as a separate entity, semantic relations between words are ignored. Thus, keyword approaches in generally do not account for query variability; they lack retrieval accuracy, and often perform a poor ranking of result sets. They do not offer meaningful ways to refine a search, for example, in cases where the number of hits exceeds any reasonable limit, nor do they offer other interactive forms of retrieval. A query is either successful or unsuccessful. When it's unsuccessful, the user has to figure out how to rephrase the query until the desired result is obtained.

### 4.2 Vector-Space Retrieval Models

Vector-space retrieval models map documents to points in a high-dimensional space where each dimension corresponds to a particular term. Then similarities between documents, as well as between documents and a query, are computed by measuring the distance between points in this space. Sophisticated term weighting schemes have been developed to give larger weight to terms that are assumed to be more important, i.e., are assumed to be more indicative of topic.

Vector-space models can handle partial matches, i.e. instances where no document exactly matches the query. They also make good use of the frequency information for result set ranking. However, vector-space models are conceptually similar to keyword-based approaches in that they ignore the semantic dimension of words. Vector-space systems, therefore, are not able to differentiate between different meanings of words—“Sun” the
celestial object vs. “Sun” the computer company—nor are they able to identify words that occur in a common context. A query for “plane” using a vector-space system would miss information referring to “aircraft”, “avion”, or “747”. Failing to take the semantic dimension of words into account results in both the introduction of substantial noise into retrieval results and the system failing to find potentially large numbers of related documents.

4.3 Concept-Based Retrieval

The insufficiencies of keyword-based retrieval have been known for more than 40 years. Concept-based retrieval systems index documents by using concepts instead of single words or phrases. While this is potentially a promising route to improve retrieval accuracy, most concept-based retrieval systems rely on linguistic resources, typically thesauri or semantic networks that have to be created and maintained by human experts. These approaches have only been moderately successful in very specialized domains, for example, the medical domain. Concepts are typically used for indexing documents as well as for normalizing and expanding queries.

There are two drawbacks of the traditional concept-based approach. The first one mentioned earlier has to do with the extremely high manual costs and expanded time requirements. Even if domain-specific thesauri are available, they are still unlikely to reflect the particular terms and concepts contained within a proprietary document collection resulting in the need for further manual effort. The second drawback is that thesauri only deal with synonyms, but synonyms are only one type of semantic relation between words. For example, “New York” and “Manhattan” are semantically related, but they are not synonyms. How about “Catholic” and “Pope”, “stock” and “Dow”, “mayor” and “city hall”, “visa” and “immigration”, “oil” and “OPEC”? Thesauri do not come close to capturing all of the semantic relationships between words.
4.4 **PLSA (Probabilistic Latent Semantic Analysis)**

PLSA is a machine learning technique that automatically identifies and structures relevant concepts and topics from a given document collection. PLSA is a patented algorithm that performs a statistical analysis of word co-occurrences in documents and identifies repeatable contexts, topics or concepts in which a certain group of words occurs. It does not require any manual input in the form of lexicons, thesauri, or topic annotations, but is completely automatic in performing “unsupervised learning”, to use the technical term. The outcome of the learning procedure is what we call a statistical model, a compressed, quantitative description of the document collection.

The identification of concepts or topics serves two major purposes: On the one hand, it reveals the potential ambiguity of words by detecting multiple contexts in which they are used. For example, “jaguar” may refer to the animal, the automobile brand, and any number of clubs, products, and businesses. “Java” might refer to the Indonesian island, the programming language, or coffee. Such ambiguities, also called polysemies, are automatically identified by PLSA whenever they are present in the source documents.

On the other hand, PLSA learns about synonyms and semantically related words, i.e., words that are likely to occur in a common context. For example, a document containing the term “car” is likely to contain synonyms like “automobile”, “auto”, “vehicle”, as well as semantically related words like “sedan”, “driving”, “highway”, and “motor”.

As opposed to other linguistic approaches that are based on lexical semantics, PLSA does not need a language-specific (or even domain-specific) thesaurus or dictionary, but learns directly from the unstructured content. This has several key advantages: First, it ensures that PLSA is applicable to any language, as long as the language can be tokenized. Second, the extracted concepts are specific to the given document population and are adapted to the language, technical terms, and specific jargon. Obviously, rebuilding similar thesauri by hand for each domain would be prohibitively expensive and time-consuming. Third, PLSA also learns a
“numerical” model, where each word has some probability to occur in a certain concept. This allows us to quantify the relationships between words. There are no thesauri or linguistic resources that provide such quantitative information.

On document level, the statistical analysis performed by PLSA has two benefits. First of all, it allows indexing documents by topic and not just by keywords. In a sense, PLSA is performing an unsupervised document categorization, using categories or topics that are suggested by the document collection. No input from a knowledge engineer or an end user is required. These topics cannot be directly observed, but are hidden (“latent”) in the text. The concept-based indexing has clear advantages over keyword-based indexing, which is the predominant paradigm for virtually all search engines and categorization systems. Most importantly, it provides a semantic representation of documents in a concept space. The semantic representation allows CORE to distinguish between true matches and documents that accidentally happen to contain some query terms but are nevertheless off topic. By working with the underlying meaning of the query, CORE presents the search results in a more meaningful way and achieves higher retrieval accuracy.

Secondly, PLSA utilizes the model to estimate the probability that a certain word will be used as a query term for a document. This will include words that did not explicitly occur in a document, but are semantically implied by related words that did occur. For example, a document containing the terms “car”, “accident”, “traffic”, etc. may not contain the word “automobile”. Nevertheless, it would obviously be a good match for a query “automobile”. Since PLSA identifies the “car” topic, it can associate related terms like “automobile” with the document. Hence, the document will have a high probability to be relevant to a query like “automobile”.

PLSA is the technology foundation upon which Recommind’s CORE™ (Content Optimized Relevance Engine) technology is built. As a result, CORE achieves an unprecedented robustness with respect to variations in the queries, and queries like “car” and “automobile”
will not return vastly different search results. Moreover, since PLSA computes actual probabilities, it is possible to rank documents according to a relevance score that provides a meaningful measure of how well a document matches an information need expressed in a query.

4.5 Advantages of CORE’s PLSA-based Approach

As described above, retrieval systems in the past have been forced to trade off the automation of simple keyword-based systems with the better accuracy of expensive, hard to maintain, and very niche linguistic approaches. CORE allows Recommind’s retrieval systems to move beyond that tradeoff and offer both the highest accuracy available with complete automation. In particular, a CORE approach offers the following technical benefits:

- In contrast to keyword-based approaches and vector space models, the retrieval engine makes use of concepts to index documents. This increases both retrieval accuracy and robustness. Queries can be matched much more reliably with content.

- Concepts and topics are automatically extracted from data in an offline process, but can be enriched and combined with existing thesauri and taxonomies, wherever available. Thus, CORE combines the scalability and degree of automation of search engines with the sophistication of concept-based retrieval models.

- Concepts identified by CORE are not purely discrete, but have a probabilistic interpretation. Further, concepts are part of a statistical model that can be used to compute a probabilistic relevance score. Bayesian inference replaces ad hoc schemes for result set ranking.

- The lexical semantics learned by CORE include different types of semantic relations and extends far beyond synonyms.
- Search results for queries that are inherently ambiguous can be organized in folders, each folder corresponding to a different concept. This is particularly helpful for queries consisting of a single search term with multiple meanings. By “disambiguating” a query, it allows users to better understand what they are looking for.

- Identifying relevant concepts also enables interactive query refinement, by suggesting semantically related terms to be included in the query. Often the user does not know how to exactly formulate a query and can zoom-in on the topic of interest, by augmenting the initial query step by step.

- When combined with categorization (described in the next section), information retrieval benefits by using the category information to refine search.

5 **Categorization Technologies**

Better browsing and routing of information

Categorization technologies are used for the filing, routing, and labeling of documents and text. Processes utilizing pure categorization, while less numerous than those utilizing information retrieval capabilities, can be very time consuming and resource-intense. These common approaches to categorization are discussed below.

5.1 **Manual Categorization**

In manual categorization, a team of subject experts goes through a collection of documents by hand and assigns each to a category or categories. This is the traditional approach to categorization and generally results in very accurate results.

In practice, however, accuracy is significantly lower than the ideal as organizations must balance the time and expense of manual categorization with the quality of the results.
Take the example of the Library of Congress, the authority when it comes to assigning subject categories to books for the United States. Here’s an example classification: the book Oliver Twist is assigned to the following subjects: 1) Orphans, 2) Criminals, and 3) London (England). The Library of Congress strives for full categorization assigning all applicable categories to a book, five or more in some cases. However, because of budget and time constraints, new books now receive a shallow categorization of only two subjects.

In addition to time and budget constraints, manual categorization is inconsistent. In study after study, domain experts disagree on the primary category for a document 50% of the time. Furthermore, the manual process is not scalable. As the number of documents and categories grow, the effort to manually tag documents expands, consuming exponentially more resources in the process.

### 5.2 Rule-Based Categorization

The first step beyond manual categorization is to construct a set of rules for categorizing documents. Typically, these systems categorize documents based on keywords in the title or document.

The drawbacks of a rule-based approach are three-fold: 1) rule-based systems require significant manual development and testing; 2) require substantial effort as new topics arise; and 3) typically have limited coverage and, as a result, limited accuracy. In summary, rule-based categorization is anything but a one-time operation as it requires constant maintenance and supervision to achieve acceptable results in a highly structured environment. In a less defined environment rules-based systems break down completely.

### 5.3 Naïve Bayes Categorization

Naïve Bayes categorization is a statistical technique for analyzing a document set and automatically assigning documents to classes (categories). The strength of Naïve Bayes is that
it is able to learn from examples. In an ideal situation, it will identify terms that have a high probability to occur in documents belonging to a category. These probabilities are then used to discriminate between different categories. Unfortunately, the real world is far from an ideal and typical application where *50-100 or even more manually labeled examples are required per category to ensure that the resulting categories are of high quality*.

Thus, the drawback of Naïve Bayes is that it requires too many examples. There is no way to begin categorizing until you have set up hundreds or even thousands of examples from which to train the system. For large organizations, with many categories and large document collections, setting up a Naïve Bayes system can engulf many months of effort to initialize and train the system, with a significant commitment to ongoing maintenance as documents and categories are added. Systems deploying personalized content delivery and agent-based information filtering, where a single user has to provide the set of training documents, represent a similar drain not just on IT resources, but on end-users themselves. In particular, during times when organizations are looking to do more with fewer resources, systems relaying on Naïve Bayes categorization are untenable.

### 5.4 Nearest Neighbor Classification

k-Nearest neighbor classification (kNN) categorizes a document based on known labels of similar documents in a training set. The majority category of the k most similar documents determines the categorization of the new document. Like Naïve Bayes, kNN often exhibits a satisfying accuracy whenever many manually labeled examples are available, but performs poorly with small and mid-sized sets of examples.

Moreover, kNN relies on a simplistic method to quantitatively determine the similarity between documents. This is typically done by counting common terms: the more words and phrases occur in both documents, the more similar they are judged. Yet, similarity measures of this type fail to represent the characteristics of a particular category. By focusing on “pairwise similarities” (similarities of pairs of documents), kNN is unable to learn what the set
of documents in a category really has in common. This makes nearest neighbor classification a poor candidate for most real-world applications.

### 5.5 Support Vector Machine Categorization

Support Vector Machines (SVM) categorization provides an excellent technology for yes/no classifications, and, as such are best suited for filters. For example, many large internet service providers use SVM technology to detect and block adult content.

SVM takes a set of positive and negative examples of a single category for training, maps these examples into an internal high-dimensional representation, and then computes linear functions on this internal representation to model the training examples. Once SVM is trained, this model categorizes new documents as belonging or not belonging to the category of interest.

SVM works well, even for small numbers of yes/no categories. However, SVM does not handle relationships between categories, or fine-grained distinctions within categories as it is typically used to construct narrow non-overlapping filters.

### 5.6 PLSA

Probabilistic Latent Semantic Analysis (PLSA) performs document categorization by automatically detecting concepts within documents via a statistical analysis of word contexts. These word contexts reflect the different concepts contained within a whole corpus of documents. Systems utilizing PLSA are then able to group documents together based on their containing similar concepts, and can do so even in the absence of taxonomies and other category information.

The PLSA algorithm is described in more detail above in Section 4.4 above. In contrast to Naïve Bayes and SVM approaches, PLSA does not require taxonomies or manually tagged documents to use as training examples. In fact, PLSA can be utilized to discover and refine
categories and taxonomies contained latently within a corpus of text. Hence, PLSA can be used stand-alone to perform text categorization or in conjunction with an approach based on a technology like Support Vector Machines depending upon the use case.

6 Personalization and Collaborative Filtering

Taking the user's interests into account

While the importance of accuracy has been emphasized throughout this paper as the driver of productivity in information retrieval and categorization systems, an important point has been left implicit until this section. That is that accuracy is difficult to measure without understanding how information is relevant to the organization or user. The importance of relevance to the task at hand is not reflected in the approaches of many earlier information technologies due to limitations in the systems available. Yet, a diplomatic organization will have an interest in the political situation in Chad the country, not in the baseball prowess of Chad the boy. By accounting for the interests of a user's personal and organizational interests, the quality and relevance of results can be much improved and the speed and efficiency with which users find information is greatly enhanced.

PLSA technology is able to seamlessly integrate user information into all aspects of search and categorization. Personalization works by seeding the statistical models generated for use in retrieval or categorization with weights appropriate for the individual or group. These can be arrived at explicitly in an initialization or implicitly using information about the user's interests garnered from previous searches and viewed documents. It can thus be deployed in a manner requiring no effort on the part of the end user.

Built-In Expert Recommendations

Most companies understand that tacit knowledge, the knowledge in peoples' minds, is the most valuable knowledge in an organization. Unfortunately, most search and categorization
systems fail to use this information. By utilizing the previous searches made by other people with whom a given individual shares common characteristics PLSA systems like Recommind’s CORE can capture not just individual preferences, but also identify groups of users with common interests. The identification of these groups enables information about what similar users found useful to be reflected in the delivery of results for queries, the prioritization of document categories, or the recommendation of related information.

7 Performance Comparison: Algorithms

Performance of information system algorithms, as described earlier, needs to be evaluated on several axes: 1) accuracy; 2) automation; and 3) usability. While clear metrics have not been established for end-user performance, academic research uses two measures, “precision” and “recall”, for evaluating accuracy of retrieval results. Precision represents the number of correct hits in a return set of specified length; and, recall measures the number of correct returns relative to the total number of possible correct returns. While these measures ignore usability in terms of time to find desired information, ease of finding desired information, etc., they provide a common (if limited) framework with which to evaluate information management systems.

7.1 Retrieval Performance

In order to provide a sense of the accuracy achieved by common systems, the accuracy of the most accurate research vector space algorithm, the Robertson TFIDF algorithm, was compared against that of PLSA. PLSA clearly demonstrates a significant precision improvement for any given recall level. The mean average precision gain over the Robertson TFIDF is 20.1%.²

² Performed on the SJMN portion of the TREC 3 data set
Of particular note is the performance of PLSA vs the vector-space algorithm at higher recall levels. Where achieving high recall (i.e. many, if not all related documents to a query) is important, the PLSA advantage is even more pronounced.

The measures of accuracy described above provide a clear demonstration of the performance advantages achieved by PLSA-based retrieval systems with respect to accuracy. While usability and automation are not factored into these measures, this omission in fact understates the performance of PLSA-based systems, which demonstrate substantive advantages in terms of both automation and usability.
Conclusion

As outlined throughout this document, the effectiveness of information retrieval and categorization systems will be determined by: 1) the work task; 2) the accuracy of the system; 3) the degree of automation in indexing, training, and ongoing maintenance that can be achieved; and 4) ease of integration with existing systems. These criteria assume that any system regardless of technology is capable of meeting basic product feature requirements with respect to functionality, scalability, and input methods. These product capabilities are discussed in detail in Recommind’s CORE technology and product literature.

Assuming these base capabilities, however, the previous discussion should have illuminated some clear trends and advantages. Not surprisingly, recent developments in machine learning and other areas of computer science have provided a more substantial theoretical starting point for development of information retrieval and categorization technologies. In particular, recent advances in statistical methods (PLSA) and other mathematical tools (SVMs) have enabled breakthrough-level quality of results. Adding to that the flexibility of deployment enabled by the self-training and category discovery of PLSA systems and a new generation of previously unobtainable productivity improvements are enabled.
Bibliography

Thomas Hofmann

Unsupervised Learning by Probabilistic Latent Semantic Analysis


Kristina Toutanova and Francine Chen and Kris Popat and Thomas Hofmann

Text Classification in a Hierarchical Mixture Model for Small Training Sets


Thomas Hofmann and Jan Puzicha

Latent Class Models for Collaborative Filtering


Thomas Hofmann

Probabilistic Latent Semantic Analysis


Thomas Hofmann

The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data


Thomas Hofmann and Joachim M. Buhmann

Active Data Clustering