# Using Taxonomies for Product Recommendation

Osvaldo Matos-Junior[1,5], Nivio Ziviani[1,4], Fabiano Botelho[3],
Marco Cristo[2], Anísio Lacerda[1,4], Altigran Soares da Silva[2,4]

[1] Universidade Federal de Minas Gerais, Brazil
{osvaldoj,nivio,anisio}@dcc.ufmg.br
[2] Universidade Federal do Amazonas, Brazil
{alti,marco}@dcc.ufam.edu.br
[3] Data Domain an EMC company, USA
fabiano@datadomain.com
[4] Zunnit Technologies, Brazil
{nivio,anisio}@zunnit.com
[5] JusBrasil, Brazil
tupy@jusbrasil.com.br

**Abstract.** In this work we take advantage of valuable information encoded in taxonomies to improve the quality of recommender systems. We present three strategies that explore the use of taxonomies: (i) category descriptors, (ii) classification features and (iii) category filters. We provide a real-case study over the book domain, in which the recommendation target is a set of 100 news page from The New York Times and the items to be recommended are 1,499,792 books distributed in 1,621 category nodes from a taxonomy, both crawled from Amazon.com. In strategy (i), term descriptors of each category are combined with text descriptions of the books assigned to the category and terms that are representative of the category are added to the target page. In strategy (ii), categories that are strongly related to the target page are put together by a classifier that plays the role of a feature generator and these features are then used in the recommendation process. In strategy (iii), the output of the two strategies previously described are filtered so that only books from the same categories as the ones assigned to the target page are kept in it. We implement several methods that apply the three strategies individually and in combination. Experimental results indicate that our strategies can be successfully applied to improving traditional content-based recommender systems. In particular, when the target page is automatically assigned to a category, we obtain gains close to 13% in average precision. On the other hand, if such an assignment is made a priori, e.g., by the author or by a content editor, the gains are close to 20% in average precision.

Categories and Subject Descriptors: H. Information Systems [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

Keywords: Recommender systems, Product recommendation, Taxonomies

## 1. INTRODUCTION

Content-based recommender systems have been successfully used in a variety of applications, ranging from recommending books, web pages, news articles, restaurants, television programs, among others [Adomavicius and Tuzhilin 2005]. In such systems an item is represented by a set of features used to describe its content. In a system to recommend books, for instance, these features correspond to a set of words, which, depending on the application scenario, may be extracted from the whole content of a book, from a summary of its content, from a description of its subject, from its title, etc. The *target* of the recommendation is a user whose information needs can be represented, for instance, by

This work was partially sponsored by the Brazilian National Institute of Science and Technology for the Web (grant MCT/CNPq 573871/2008-6), by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, and authors individual grants and scholarships from CNPq.

a profile, by another book, or by a web page. Targets are represented by features of the same type as the ones used to represent the items, i.e., words in our example. By comparing new items with the target, it is possible to determine the items that are most relevant to it.

The problem with this approach is that a same concept or semantic relationship can be described using different word sets. For instance, the target description might include the word "soccer" whereas a description of a book about soccer would miss the word "soccer" but includes the synonym "football" or the related word "league". Thus, the simple matching of the terms used to describe items may not be enough to evaluate similarity. A possible strategy to deal with this problem is to resort to knowledge obtained from external sources, i.e., sources other than the target and the items.

In this work we investigate how to take advantage of information encoded in a *taxonomy* to deal with the mismatch problem. Using taxonomies and other human-constructed knowledge bases opens the opportunity to incorporate domain-specific and common-sense knowledge compiled by humans. Such source of information could not be obtained otherwise from neither the target nor the items alone. Furthermore, taxonomies are currently a common resource maintained by companies that operate electronic commerce systems which offer recommender services.

Our investigation is based on a case study over the book domain. The target is a set of 100 news page from The New York Times and the items to be recommended are 1,499,792 books distributed in 1,621 category nodes from a taxonomy, both crawled from Amazon.com. We argue that a system to recommend books is a representative real-case application that provides an adequate context to experiment with distinct strategies. In fact, having such a rich source of information publicly available for experimentation was one of the reasons why we chose the book domain to conduct this case study.

Within this scenario, we present three distinct strategies to exploit the use of taxonomies in content-based recommender systems: (i) *category descriptors*, in which terms that are representative of a category are added to the representation of the target page; (ii) *classification features*, in which categories that are strongly related to the target page are used as features in the recommendation process; and (iii) *category filters*, in which only items associated to categories considered as related to the target page are kept in the recommendation results. Besides proposing these strategies, we implemented several methods for content-based recommendation that apply them individually and in combination, and perform a comprehensive set of experiments using these methods.

Our strategies consider two distinct scenarios regarding the assignment of the target page to one or more categories of a taxonomy: (i) an automatic classifier does this assignment without user intervention and (ii) the target page is assigned beforehand to one or more categories, e.g., by editors of articles of an on-line newspaper that recommends books to readers. Experimental results indicate that, when the target page is automatically assigned to a category, we obtain gains close to 13% in average precision. On the other hand, if such an assignment is made a priori, the gains are close to 20% in average precision.

The remaining of this article is organized as follows. In Section 2 we discuss related work. In Section 3 we introduce our strategies to improve current recommendation systems by adding information obtained from taxonomies. In Sections 4 through 6 these strategies are described in detail. In Section 7, we present our experimental evaluation, which includes the description of datasets, evaluation metrics, and results. Finally, in Section 8, we present our conclusions and future research directions.

## 2. RELATED WORK

Previous works in the literature have already exploited taxonomies to address several information retrieval tasks. In the context of web search, improvements can be obtained by expanding queries with terms related to the entire corpus (global expansion), to a set of documents similar to the query (local expansion), or to the query category (category-based expansion). While some of the metrics we

use here were previously introduced by methods that employ local expansion [Carpineto et al. 2001; Carpineto and Romano 1999], the methods most related to ours are those that use category-based expansion. For instance, Inquirus2 [Glover et al. 2001] uses entropy loss to select new terms for an issued query. Keyword spices [Oyama et al. 2001] and TAX-PQ [Pahlevi and Kitagawa 2005] use decision trees to find new terms in documents classified according to hierarchical and flat taxonomies. Our method differs from those because we aim at enriching the representation of web pages instead of queries.

In [Gabrilovich and Markovitch 2005], the authors proposed an alternative approach to query expansion. They use the categories from a taxonomy to build an additional feature space of *classification features*, in addition to the usual bag of word features, and use these two feature spaces in combination to improve text categorization.

Regarding web advertising, this approach was later adopted in [Anagnostopoulos et al. 2007], allowing to match ads and web pages based on both bag of words features and classification features. These features were obtained from a comprehensive taxonomy with about 6,000 categories. In this work, however, the focus was not on improving ad recommendation, but on minimizing the size of the pages without sacrificing the quality of the ad selection.

In [Hung 2005] the authors use a product taxonomy to classify consumers according to their behavior. They were interested in distinguishing consumers which prefer specific products from the ones which prefer specific brands. Taxonomies were used to assist identifying general product categories.

In traditional content-based recommender systems [Adomavicius and Tuzhilin 2005], the system learns to recommend items that are similar to the ones that the user liked in the past. A collaborative system proposed in [Ziegler et al. 2008] uses a taxonomy provided by Amazon to recommend books. They were able to improve the overall user satisfaction in expense of some loss in the prediction accuracy of the utility of the items. One difference between [Ziegler et al. 2008] and our method is that they represent users as a set of book categories and we represent them as a set of books. Another difference is that they use information on ratings given by other users in the system. In our case we use just the content of the page seen by the user, without information from other users. For this reason we do not discuss traditional content-based recommender systems here.

## 3. USING TAXONOMIES

In this section we introduce three strategies to improve current content-based recommender systems by adding several new pieces of information obtained from taxonomies. Given a recommendation *target* and a collection of *items*, the content-based recommendation task consists of finding a subset of the items semantically related to the target.

Content-based recommender systems often use a matching approach in which the target and the items to be recommended are represented as *bags of words* (*BOWs*). The BOWs are used to build TF-IDF weighted term vectors [Salton and Buckley 1988] and a *ranking* of items is generated based on the similarity between each item and the target. This similarity can be computed using the cosine measure [Baeza-Yates and Ribeiro-Neto 2011] or one of its variations [Zobel and Moffat 2006]. The list of recommended items corresponds to the top-$K$ items in the ranking. The value of $K$ is usually small (around ten items) and depends on specific application scenarios.

There are many situations, however, for which the simplistic representation based on BOW leads to poor results (e.g., [Linden et al. 2003; Gabrilovich and Markovitch 2005]). This problem occurs in cases where the semantic relationship across objects to be matched cannot be properly captured by simply matching the terms used to describe them. This phenomenon is the *vocabulary impedance problem* coined in [Ribeiro-Neto et al. 2005]. A common strategy to deal with such a problem is to enrich the representation of the objects to be matched using additional sources of information [Ribeiro-Neto et al.

2005; Carpineto et al. 2001; Carpineto and Romano 1999]. We experiment this same general idea, but, instead of using related pages, we rely on information obtained from categories of a taxonomy.

Using taxonomies opens the opportunity to incorporate domain-specific and common-sense knowledge compiled by humans, something that could not be obtained otherwise from neither the target nor the items alone. We use 1,621 nodes distributed in seven levels derived from the taxonomy made available by *Amazon.com* through its Amazon Web Services (AWS[1]). Our case study also used information on 1,499,792 books available from AWS. We collected the textual description of each book and the categories each book has been assigned to in the book taxonomy (there can be more than one category per book).

Figure 1 illustrates a framework for content-based recommendation, in which the recommendation target is a news web page and the items to be recommended are books. Web pages are represented by their textual contents and books are represented by textual descriptions of their contents. The three strategies we propose, which are based on using category descriptors, classification features and category filters, are also illustrated. In the following sections we present a detailed description of each strategy.

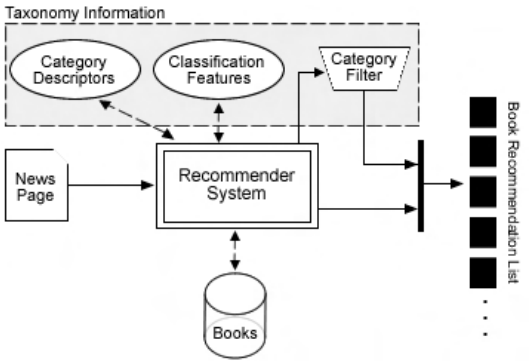


Fig. 1. An extended framework for content-based recommendation.

## 4. CATEGORY DESCRIPTORS

In this section we describe our first strategy, which is based on using terms relevant to a category to enrich the representation of target pages. The process we use consists in: (1) assigning the target page to one or more categories in the taxonomy; (2) taking the textual descriptions of the books assigned to these categories and selecting from them a number of *descriptors*, i.e., terms that are representative of the category; (3) adding these descriptors to the representation of the target page by using linear combination.

For the first step, two distinct scenarios are considered: (i) an automatic classifier does this assignment without user intervention and (ii) the target page is assigned beforehand to one or more categories, e.g., by the author of the target page or by a content editor. We evaluate both scenarios in our experiments. As in both scenarios the assignment of pages to categories can be carried out by using standard procedures, we do not further elaborate on it. The second step, though, requires a more detailed discussion, which we present in the following.

Consider a category $C$ in the taxonomy and let $T(C)$ be the set of all terms occurring in the descriptions of the books assigned to this category. We look for a set of terms $D(C) \subseteq T(C)$ that are representative of category $C$, i.e, the *Category Descriptors*. For this, we use some function $m(t, C)$

[1] `http://aws.amazon.com/`

that tries to assess the importance of term $t$ to category $C$ and build a set $D^K(C)$ of the top-$K$ descriptors, i.e., the $K$ terms that yield to the highest values of $m(t, C)$. In our work we experiment with four distinct measures for implementing this function.

The measures we will present next use the following notation: (i) $P(t)$ is the probability of a given term $t$ in the collection of books and $P(t|C)$ is the probability of term $t$ given a category $C$. Both probabilities are estimated by the ratio of book descriptions containing the term $t$ over all books or in the category; (ii) $B_t$ is the set of all book descriptions containing a term $t$ and $B_C$ is the set of descriptions of all books assigned to a given category $C$.

*Kullback-Leibler Divergence (KLD).* Or *relative entropy*, it estimates the difference between two probability mass functions $p(x)$ and $q(x)$ — the distance between two probability distributions [Kullback and Leibler 1951; Carpineto et al. 2001]. In our case, $p(x)$ represents the observations of $P(t|C)$, and $q(x)$ is $P(t)$. Hence, it is defined as:$P(t|C) \times \log \frac{P(t|C)}{P(t)}$.

*Pearson's Chi-Squared (CHI2).* This is a statistical test used for computing the relationship between an expected frequency in the general population and an observed frequency [Carpineto et al. 2001; Croft et al. 2009]. In our work, the expected frequency is the term likelihood $P(t)$, and the observed frequency is the probability of $t$ conditioned on the category $C$, i.e., $P(t|C)$. Hence, it is defined as: $\frac{(P(t|C)-P(t))^2}{P(t)}$.

*Dice's coefficient (DICE).* It measures the similarity between two sets. For instance, in [van Rijsbergen 1979; Croft et al. 2009] it is used as a term association measure. In our case, we use this measure to evaluate the similarity between $B_t$ and $B_C$. If these two sets are similar, then $t$ is considered as closely related to $C$. In this case, the Dice's coefficient is given by: $\frac{2\times|B_t \cap B_C|}{|B_t|+|B_C|}$.

*Document Frequency (DF).* We also experiment with a simple measure based on the term frequency within a category, which corresponds to the intersection between $B_t$ and $B_C$, i.e., $|B_t \cap B_C|$.

To illustrate the use of these measures, we present in Table I the top-10 category descriptors obtained by each of them for the category *Religion*, when considering our taxonomy and the companion book descriptions. Notice that, in comparison to the other measures, DF also selects general terms from the whole collection, such as book, world, work, etc. An additional measure we refer to as ALL is also presented in this table, which is calculated by averaging the values obtained with KLD, CHI2 and DICE for a given term. Before averaging these values, we first normalize them to obtain values between 0 (lower score) and 1 (highest score). This measure is also used in our experiments. We notice that the measures KLD, CHI2 and DICE, are commonly used for query expansion (e.g, in [Croft et al. 2009; Carpineto et al. 2001]).

Table I. Top-10 descriptors for *Religion*.

| KLD | CHI2 | DICE | DF | ALL |
|---|---|---|---|---|
| god | god | god | book | god |
| christian | christian | spiritual | life | christian |
| spiritual | spiritual | christian | god | spiritual |
| bible | bible | life | new | bible |
| church | church | bible | world | church |
| faith | jesus | church | spiritual | faith |
| life | christ | faith | people | jesus |
| jesus | faith | book | time | christ |
| christ | biblical | religious | work | life |
| religious | christians | jesus | christian | biblical |

For the recommendation process, we add the category descriptors to the BOW representation of the target page. Further, the score obtained for a given term $t$ through any of the aforementioned measures is also used to recompute its weight. For this, we resort to a linear combination similar to the one used in the Rocchio's formula [Rocchio 1971].

Consider $t$ as a term in the extended BOW representation of the target page. The weight of this term is computed as:

$$w'_{t,p} = (1 - \alpha) \times w_{t,p} + \alpha \times m(t, C) \quad (1)$$

where $\alpha$ is a positive constant between 0 and 1, $w_{t,p}$ is the weight of $t$ in the target page $p$ computed according to the traditional TF-IDF weighting scheme [Salton 1989] and $m(t, C)$ is the value obtained for $t$ in the category $C$ to which $p$ was assigned, using one of the measures previously described. To avoid discrepancies, we first normalize the components of this formula to the same scale. When $t$ is a descriptor but is not present in the page $p$ we zero out $w_{t,p}$. Otherwise, when $t$ is in the page $p$ but is not a descriptor we zero out $m(t, C)$.

Notice that if the target page is assigned to several categories, we calculate $w'_{t,p}$ by averaging the scores $m(t, C)$ obtained for all categories. We experiment this scenario when using an automatic classifier to predict the page categories.

In practice, the new representation of the target page $p$ may include a very large number of terms because of the addition of the category descriptors. However, previous experiments we have conducted show that most of these terms do not have a significant impact on the representation, since they have a very low weight. Therefore, we prune this representation by keeping only the $N$ highest-weighted terms, where $N$ is equal to the number of terms in the original page representation.

## 5. CLASSIFICATION FEATURES

In this section we describe how to improve the simple BOW approach by using, as in [Gabrilovich and Markovitch 2005], an additional feature space of *classification features* built using categories of a taxonomy.

In this strategy, each target page $p$ is represented by a vector $\vec{p}$ in the term space plus another vector $\vec{p}_{cat}$ in the category space. This category vector is built using the set of categories predicted by a *classifier*. Thus, these categories are assumed to be strongly related to the page. In our case, a centroid-based classification approach [Han and Karypis 2000] is used, which plays the role of a feature generator.

In the following, we first show how category centroids are computed. Next, we present the procedure we use to generate the classification features for a target page $p$. Finally, we describe how these features are incorporated in the recommendation process.

Prior to the generation of the classification features, a term vector $\vec{c}_j$ has to be generated for each category $C_j$ in the taxonomy. This vector corresponds to the centroid of the term vectors generated for the books assigned to the category.

To compute $\vec{c}_j$, a BOW is built for each book $b$ assigned to $C_j$ using the terms available in its title and description. Stop-words, numbers, mixed words (words with digits) and very rare words (i.e., those available in no more than five items) are removed.

A term vector $\vec{b}$ is then generated for book $b$ using these stems according to the vector-space model. For this, we consider that the vocabulary of the collection corresponds to the term space.

Let $B_j$ be the set of vectors that represent the books assigned to category $C_j$. The centroid vector $\vec{c}_j$ representing $C_j$ is defined as:

$$\vec{c_j} = \frac{1}{|B_j|} \sum_{\vec{b} \in B_j} \vec{b} \tag{2}$$

Following [Gabrilovich and Markovitch 2005], for our experiments we build $\vec{c_j}$ using only the 1,000 more frequent terms in the category.

Our procedure for generating the classification features for a page $p$ is described by Algorithm 1 and detailed in the following.

**Algorithm 1** Generation of Classification Features.

**Input:** target page $p$
**Output:** category vector $\vec{p}_{cat}$ for $p$
1: **for all** categories $C_j$ in the taxonomy **do**
2: $\vec{p}_{cat}[C_j] \leftarrow 0$
3: $S \leftarrow segments(p)$
4: **for all** segments $s \in S$ **do**
5: Generate a term vector $\vec{s}$ for $s$
6: **for all** categories $C_j$ in the taxonomy **do**
7: **let** $\vec{c_j}$ be the centroid vector generated for $C_j$
8: $sim(s, C_j) \leftarrow \cos(\vec{s}, \vec{c_j})$
9: $\mathcal{C}^K \leftarrow$ the $K$ categories most similar to $s$
10: **for all** $C_j \in \mathcal{C}^K$ **do**
11: $\vec{p}_{cat}[C_j] \leftarrow \vec{p}_{cat}[C_j] + sim(s, C_j)$
12: **let** $\mathcal{A}$ be the set of ancestors of the categories in $\mathcal{C}^K$
13: **for all** $C_\ell \in \mathcal{A}$ **do**
14: $\vec{p}_{cat}[C_\ell] \leftarrow \vec{p}_{cat}[C_\ell] + sim(s, C_\ell) \cdot \epsilon$

After an initialization step, in Line 3, the algorithm extracts a number of segments from the target page $p$. The goal here is to be able of selecting categories related to the whole page based on the categories that are related with each segment from the page. In our experiments we have tried several distinct segmentation schemes, as suggested in [Gabrilovich and Markovitch 2005]. However, two of them yielded the best results: (i) taking the entire content of a page as a single segment, and (ii) *multi-resolution* approach [Gabrilovich and Markovitch 2005], i.e., windows of words (bigrams), paragraph and the entire document content.

The algorithm then iterates through the segment (Loop 4–14). For each segment $s$, a term vector $\vec{s}$ is generated using the standard TF-IDF weighting scheme (Line 5), so that it is possible to determine the similarity between the segment and each category by calculating the cosine of the angle between vectors $\vec{s}$ and $\vec{c_j}$ (Line 8).

In Lines 10–11 the category vector $\vec{p}_{cat}$ is built using the $K$ most relevant categories to each segment. Notice that the weight of each category depends on the number of segments it is related to and on the similarity scores between the segment and the category. Additionally, in Lines 13–14, the ancestors of these top-$K$ categories in the taxonomy are also included as classification features with a weight amortized by a constant $\epsilon$. This aims at capturing higher-level concepts (e.g., sports in addition to soccer), avoiding, however, their dominance on the vector. Similarly to what is done in [Anagnostopoulos et al. 2007], in our experiments we use $K = 5$ and $\epsilon = 0.5$.

The classification features generated for the target page $p$ using Algorithm 1 can now be used for the recommendation process. For this, the term vector $\vec{p}$ in the term space is used along with the vector $\vec{p}_{cat}$ in the category space to match the target page $p$ and a book $b$ as follows:

$$sim_{CLF}(p, b) = \alpha \cdot \cos(\vec{p}, \vec{b}) + \beta \cdot \cos(\vec{p}_{cat}, \vec{b}_{cat}) \tag{3}$$

where $\vec{b}$ and $\vec{b}_{cat}$ are respectively the term vector and category vector for $b$, and $\alpha$ and $\beta$ are constants (in our experiments we used $\alpha = \beta = 1$).

The vector $\vec{b}$ is generated according to the vector-space model using the BOW representation of book $b$. In the case of $\vec{b}_{cat}$, we recall that all books are associated beforehand to a set of categories from the Amazon taxonomy. We take advantage of this high-quality information and build this vector by assigning 1 to the dimensions corresponding to each of these categories.

## 6. CATEGORY FILTERS

The two strategies previously described are used to improve the matching between the target and the items to be recommended. Our third strategy is based on a *Category Filter* that works on the output of that matching, thus being independent from any particular matching strategy.

In our case study, it works as follows. First, the target page is mapped to one or more categories of the book taxonomy. Next, the list of recommended books is obtained using the BOW representation, possibly augmented with the application of the previously described strategies. This list is then filtered so that only books from the same categories as the ones assigned to the target page are kept.

The main idea behind the category filter strategy is the following: the strategy of augmenting the representation of the page with category descriptors effectively favors recall, since the number of matching books is likely to increase. This, however, might result in a drift off the target page topic. The category filter prevents the drift by pruning out-of-topic results, thus favoring precision.

For this strategy to work properly, we assume that the target web page is assigned to the appropriate categories in the taxonomy. In our work we experiment with both scenarios, i.e., when an automatic classifier does this assignment and when the assigned is made beforehand, e.g., by the editor of an on-line newspaper.

## 7. EXPERIMENTAL EVALUATION

To evaluate our strategies described in Section 4 through 6, we implement several methods for content-based recommendation that apply them individually and in combination. Using these methods we perform a comprehensive set of experiments with a representative collection of target web pages. In the following we describe the datasets and the metrics used in these experiments, present the implemented methods and finally discuss the experimental results.

### 7.1 Datasets

To perform our experiments we use: (1) a collection of 100 news pages from The New York Times (NYT)[2] crawled on November 2009, with information of the topic a page belongs to. Each page was used as a recommendation target in our experiments; (2) a collection of books crawled from *Amazon.com*. A set of terms was extracted from each page and submitted to the AWS (Amazon Web Service) to obtain a correspondent XML page. We first removed duplicates using the ISBN of each book and the ASIN (Amazon product identification number) code and obtained a set of 6,372,612 book descriptions, distributed in 34 categories in the level 1 of the hierarchical Amazon book taxonomy – the levels are numbered in a top-down fashion, where the level 0 is the taxonomy's root which is "Books". Next, we removed items without ISBN, product and/or category description, books in other types of media (e.g., audiobook), and any product other than book (e.g., DVD, mouse), ending up with a collection of 1,499,792 books in 28 categories in the level 1. (3) a book taxonomy extracted from the set of XML pages. For an initial set of 11,299 nodes we removed those such as "custom

[2] `http://www.nytimes.com`

stores", "feature stores" and navigation nodes, ending up with a taxonomy containing 7 levels and 1,621 category nodes. One book might fit in more than one category (i.e., multi-label case).

For the sake of reproducibility the datasets and the human evaluations described later on are available at http://www.latin.dcc.ufmg.br/collections/jidm2012/.

As mentioned earlier, our strategies require one or more categories to be associated with the target page. As already mentioned, we consider two scenarios: (i) when an automatic classifier does this assignment, and, (ii) when the assigned is made beforehand by the author of the target page or a content editor. For the first scenario, although we could have used any automatic text classifier, we adopted the same centroid-based classifier described in Section 5. For the second scenario, we simulate the role of a content editor and associate each of the 100 target news page to one category in the book taxonomy. This association was made a priori based on human-judgment on which category best represents the main subject of the page.

### 7.2 Evaluation Metrics

For the evaluation of each recommendation method we use a pooling protocol similar to what was done for TREC [Voorhees and Harman 1998]. In our experiments, each NYT news page corresponds to a topic and the recommendation methods generated runs with a ranked list of recommended books for different topics. We considered the 100 topics from the NYT collection as input to different recommendation methods and generated page-book pairs. Since the recommendation task in general involves a few items, for each run we collected the top $k$ books relevant to the topic, where $k = 5$. By removing the duplicate page-book pairs we ended up with 7,380 distinct pairs. As discussed in [Carterette et al. 2006], a collection of such a size is enough to give 95% confidence.

We used 15 annotators (volunteers from the Graduate Program in Computer Science from the Universidade Federal de Minas Gerais, Brazil) to judge the page-book pairs. For each topic we present the selected books in a random order to an annotator to mark each book as "relevant" or "non-relevant" to the topic, according to the following question: If you would be accessing this page on the Internet, which books would you recommend as "relevant" or "non-relevant" to the topic?

We then use the human judgments to evaluate how well the different recommendation methods distinguish the recommended list of books for each topic. The common pooling practice assumes that the most relevant books to each page have been judged and non-judged books are considered non-relevant. As the number of relevant books for each news page in our collection is large, the number of judged books is just a small fraction of all the books available for retrieval, and the top $k$ retrieved books might contain from 0 to $k$ judged as relevant books. As we will see, there are metrics used for retrieval evaluation with incomplete information which allow to overlook non-judged books without requiring to consider them to be non-relevant.

To compute the precision at various levels of recall (p@k and pavg@k [Baeza-Yates and Ribeiro-Neto 2011]), in our evaluation of each news page we consider only those books for which we have judgments. For the mean average precision (MAP) we considered the top 10 books because a few items are recommended at a time and therefore precision is more important than recall. The two aforementioned metrics are defined as follows:

$$p@k = \frac{\sum_{j=1}^{k} rel(j)}{k} \tag{4}$$

$$pavg@k = \frac{1}{k} \sum_{i=1}^{k} \left( rel(i) \times \left( \frac{\sum_{j=1}^{i} rel(j)}{i} \right) \right) \tag{5}$$

where $rel(i)$ is a binary function used to indicate whether the book ranked at the $i$-th position is relevant to the target page topic.

As mentioned, there are metrics that allow to overlook non-judged books without requiring to consider them to be non-relevant, such as *bpref* [Buckley and Voorhees 2004] and inferred average precision *infAP* [Yilmaz and Aslam 2006]. In this work we use the *infAP* metric instead of *bpref* because it is closer to the MAP metric [Yilmaz and Aslam 2006].

The *infAP* metric [Yilmaz and Aslam 2006] is obtained by averaging the expected precision at the rank of each relevant book using Equation 6. Relevant books that are not retrieved by the system are assumed to have precision zero. As for the MAP we also considered the top 10 books for the *infAP* metric.

$$E[\text{prec at rank } k] = \frac{1}{k} + \frac{k-1}{k} \times E[\text{prec above } k]] \tag{6}$$

where

$$E[\text{prec above } k] = \frac{\text{judged relevant above } k}{\text{judged relevant above } k + \text{judged non}-\text{relevant above } k} \tag{7}$$

In all cases, the comparison of the results obtained with our method and baseline methods is verified using the Student's t-test [Jain 1991]. Since we do not know the standard deviation for our population (all possible comparisons between our methods), we used a paired test. As our null hypothesis, we consider that the compared methods have the same performance. For the results reported in the following sections the symbol † indicates that the null hypothesis was rejected with a 95% chance while the symbol ‡ indicates that it was rejected with 99% chance.

### 7.3 Recommendation Methods

Table II presents the list of recommendation methods we have implemented based on our three strategies: DESC (category descriptors), CLF (classification features) and CTF (category filtering). The additional BOW method is a simple one that uses TF-IDF weights, as described in Section 3. It plays the role of a baseline in our evaluation.

Table II. Recommendation methods

| Abbreviation | Method |
|---|---|
| BOW | Bag of Words |
| DESC | Category Descriptors |
| DESC-CHI2 | DESC with Pearson's Chi-Squared |
| DESC-KLD | DESC with Kullback-Leibler divergence |
| DESC-DICE | DESC with Dice's coefficient |
| DESC-DF | DESC with document frequency |
| DESC-ALL | DESC with combined measures |
| CLF | Classification Features |
| CLF-EC | CLF using entire page content |
| CLF-SE | CLF using segmented page |
| CTF | Category Filtering |
| CTF-A | CTF Automatic |
| CTF-B | CTF Beforehand |

For the DESC strategy, the five measures give us the following methods: Pearson's Chi-Squared (DESC-CHI2), Kullback-Leibler divergence (DESC-KLD), Dice's coefficient (DESC-DICE), document frequency (DESC-DF), and a combination of measures (DESC-ALL). In the CLF strategy, the method CLF-EC uses the entire content of the page for the classification features and the method CLF-SE uses segments according to the *multi-resolution* approach [Gabrilovich and Markovitch 2005] (see Section 5). In the CTF strategy, the method CTF-A uses categories associated to the page by using

an automatic classifier, whereas the method CTF-B uses categories associated beforehand to the target page.

Besides the methods derived from each strategy, we also experiment with combinations of different strategies creating new hybrid methods.

### 7.4 DESC Results

In Equation 1, the influence of the category descriptors in the DESC methods is adjusted by parameter $\alpha$. Figure 2 shows the impact in the *infAP* measure by varying $\alpha$ between 0 (only terms of the page are used) and 1 (only category descriptors are used). Recall that $\alpha = 0$ means that only BOW is used. For this experiment, we considered only one category associated beforehand to the target page.

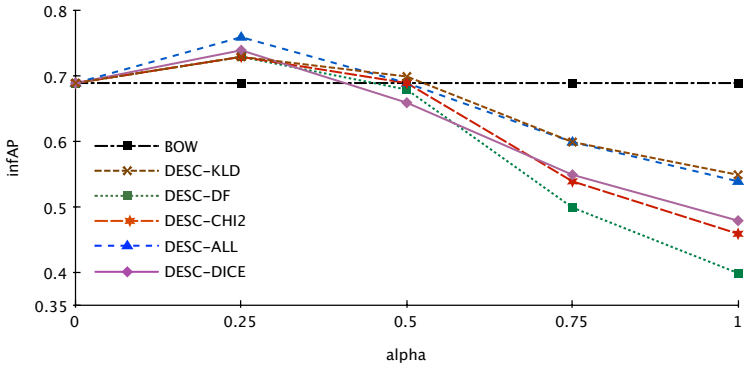


Fig. 2. Varying the value of $\alpha$.

The plots show that all the methods based on category descriptors have their best results when $\alpha = 0.25$. This means that using category descriptors can indeed improve the recommendation quality, but that it is also important to preserve the influence of terms of the page, which ultimately represent its subject. For this reason we adopt $\alpha = 0.25$ in the following experiments with category descriptors. Notice, however, that when $\alpha$ approaches 1, the more general terms of DESC-DF hurt the quality of the recommendation, but DESC-ALL and DESC-KLD keep *infAP* above 0.5.

In Table III, the method DESC-ALL, which combines different measures, outperforms all the methods using a single measure. The gain of the method DESC-ALL over the baseline BOW is approximately 20% for the metric p@1. Thus, in the same table we present results for the method DESC-ALL considering the scenario in which the categories are automatically assigned to the page. These methods are called DESC-ALL-$n$A, where $n$ is the number of categories assigned to the target page.

Table III. Precision values for the DESC strategy.

| | p@1 | p@3 | p@5 | MAP | infAP |
|---|---|---|---|---|---|
| DESC-ALL | $0.79^{\ddagger}$ | $0.72^{\ddagger}$ | $0.72^{\ddagger}$ | $0.80^{\ddagger}$ | $0.76^{\ddagger}$ |
| DESC-DICE | $0.78^{\dagger}$ | $0.71^{\ddagger}$ | $0.72^{\ddagger}$ | $0.79^{\dagger}$ | $0.75^{\ddagger}$ |
| DESC-CHI2 | $0.75^{\dagger}$ | $0.70^{\ddagger}$ | $0.70^{\ddagger}$ | $0.78^{\ddagger}$ | $0.74^{\ddagger}$ |
| DESC-DF | 0.73 | $0.72^{\ddagger}$ | $0.72^{\ddagger}$ | $0.77^{\ddagger}$ | $0.73^{\dagger}$ |
| DESC-KLD | 0.74 | $0.69^{\ddagger}$ | $0.70^{\ddagger}$ | $0.77^{\dagger}$ | $0.73^{\dagger}$ |
| DESC-ALL-5A | 0.70 | $0.71^{\ddagger}$ | $0.70^{\ddagger}$ | 0.76 | 0.72 |
| DESC-ALL-10A | 0.72 | $0.70^{\ddagger}$ | 0.67 | 0.74 | 0.70 |
| DESC-ALL-1A | 0.69 | $0.67^{\ddagger}$ | $0.68^{\dagger}$ | 0.74 | 0.70 |
| BOW | 0.66 | 0.61 | 0.62 | 0.70 | 0.67 |

We can see that methods that rely on a beforehand category assignment outperform methods that use automatic category assignment. Among the methods that use automatic category assignment, DESC-ALL-5A yields the best results. One possible explanation is that: (1) using more than one category allows for diverse concepts to be added, which contributes to a richer representation; (2) eventual classification errors, to which automatic classifiers are prone to, can be harmful to the selection of category descriptors – in the case of DESC-ALL-1A, if the single category is incorrect, only unrelated descriptors can be added, whereas in the case of DESC-ALL-10A, several of such descriptors can be added.

### 7.5 CLF Results

Figure 3 shows the results for the CLF methods in comparison with the baseline BOW. Both methods improve the recommendation results, but for the first 70% levels of recall the method CLF-EC (using the entire content of the page) is better than the CLF-SE (using segments).

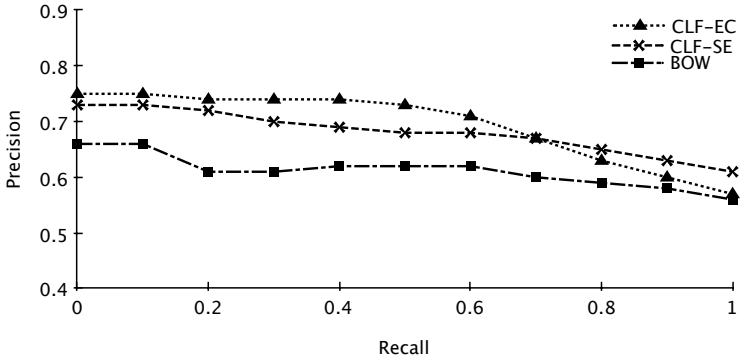


Fig. 3. Comparing BOW with CLF-EC and CLF-SE.

The precision values presented in Table IV confirm that the methods based on the generation of classification features perform better than the baseline method BOW. For instance, considering the metric p@3, the methods CLF-EC and CLF-SE overcome the baseline method in approximately 21% and 15%, respectively. As can be seen from the table, the method CLF-EC had better performance in all metrics. We stress that in CLF methods the scenario of beforehand category assignment does not apply, since the CLF strategy is based on the use of an automatic classifier.

Table IV. Precision values for the CLF strategy.

| | p@1 | p@3 | p@5 | MAP | infAP |
|---|---|---|---|---|---|
| CLF-EC | 0.75 | $0.74^{\ddagger}$ | $0.71^{\ddagger}$ | $0.78^{\ddagger}$ | $0.74^{\ddagger}$ |
| CLF-SE | 0.73 | $0.70^{\ddagger}$ | $0.68^{\ddagger}$ | $0.76^{\ddagger}$ | $0.72^{\ddagger}$ |
| BOW | 0.66 | 0.61 | 0.62 | 0.70 | 0.67 |

### 7.6 CTF Results

Figure 4 presents precision versus recall plots for methods based on the category filtering strategy. Three methods use an automatic classifier to assign categories to the target page, namely CTF-1A, CTF-5A and CTF-10A, which use one, five and ten categories, respectively; and one method, CTF-B, uses a single category assigned beforehand to the target page. The plots show that CTF-5A and CTF-10A outperform the baseline method BOW considering all levels of recall, and that the method CTF-B surpasses all methods.

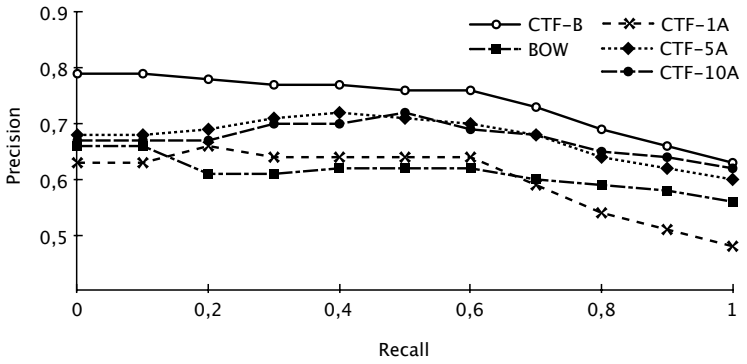


Fig. 4. Results of CTF methods.

The positive impact of the human intervention is also evident in Table V, where the method CTF-B outperforms the baseline method BOW by approximately 26% in the metric p@3. On the other hand, the best method using automatically assigned categories, CTF-5A, presents a gain of approximately 8% over the baseline.

Table V. Precision metrics for the CTF strategy.

| | p@1 | p@3 | p@5 | MAP | infAP |
|---|---|---|---|---|---|
| CTF-B | $0.79^{\ddagger}$ | $0.77^{\ddagger}$ | $0.76^{\ddagger}$ | $0.80^{\ddagger}$ | $0.76^{\ddagger}$ |
| CTF-5A | 0.68 | $0.71^{\ddagger}$ | $0.71^{\ddagger}$ | $0.76^{\dagger}$ | $0.72^{\dagger}$ |
| CTF-10A | 0.67 | $0.70^{\ddagger}$ | $0.72^{\ddagger}$ | 0.75 | 0.71 |
| CTF-1A | 0.63 | 0.64 | 0.64 | 0.69 | 0.64 |
| BOW | 0.66 | 0.61 | 0.62 | 0.70 | 0.67 |

The plots in Figure 4 and the values in Table V show that using one category filter (CTF-1A), in the automatic scenario, does not give good results and that five categories are sufficient to obtain significant improvements. As in Section 7.4, this number of categories is enough to capture the diversity of concepts from several categories but avoids the introduction of noisy due to possible misclassification.

### 7.7 Comparing Strategies

Table VI presents a comparison of the best methods for each strategy, considering automatic and beforehand category association. The lines in this table corresponding to the HYBRID methods will be discussed shortly in the next section.

Table VI. Comparison of the best methods for each recommendation strategy.

| | p@1 | gain(%) | p@3 | gain(%) | p@5 | gain(%) | pavg@3 | gain(%) | pavg@5 | gain(%) |
|---|---|---|---|---|---|---|---|---|---|---|
| HYBRID-B | $0.81^{\ddagger}$ | 22.73 | $0.78^{\ddagger}$ | 27.87 | $0.76^{\ddagger}$ | 22.58 | $0.87^{\ddagger}$ | 19.18 | $0.86^{\ddagger}$ | 19.44 |
| CTF-B | $0.79^{\ddagger}$ | 19.70 | $0.77^{\ddagger}$ | 26.23 | $0.76^{\ddagger}$ | 22.58 | $0.84^{\ddagger}$ | 15.07 | $0.83^{\ddagger}$ | 15.28 |
| DESC-ALL | $0.79^{\ddagger}$ | 19.70 | $0.72^{\ddagger}$ | 18.03 | $0.72^{\ddagger}$ | 16.13 | $0.84^{\ddagger}$ | 15.07 | $0.83^{\ddagger}$ | 15.28 |
| CLF-EC | 0.75 | 13.64 | $0.74^{\ddagger}$ | 21.31 | $0.71^{\ddagger}$ | 14.52 | $0.82^{\dagger}$ | 12.33 | $0.82^{\ddagger}$ | 13.89 |
| DESC-ALL-5A | 0.70 | 6.06 | 0.71 | 16.39 | 0.70 | 12.90 | 0.80 | 9.59 | $0.79^{\dagger}$ | 9.72 |
| CTF-5A | 0.68 | 3.03 | $0.71^{\ddagger}$ | 16.39 | $0.71^{\ddagger}$ | 14.52 | 0.77 | 5.48 | 0.78 | 8.33 |
| HYBRID-A | 0.66 | 0.00 | $0.72^{\ddagger}$ | 18.03 | $0.70^{\dagger}$ | 12.90 | 0.78 | 6.85 | 0.77 | 6.94 |
| BOW | 0.66 | - | 0.61 | - | 0.62 | - | 0.73 | - | 0.72 | - |

In the table, we first observe that, as it could be anticipated, the methods that rely on beforehand category assignment are always better than methods that use automatic assignment. Among the automatic methods, CLF-EC outperforms both DESC-ALL-5A and CTF-5A. It is worth noticing that automatic methods DESC-ALL-5A and CTF-5A, both using 5 categories, were the best among the methods within their respective strategies. Again as in Section 7.4, the use of five categories balanced diversity and noise avoidance.

Finally, we highlight that CLF-EC seems to be a very interesting option to consider if beforehand category assignment is not possible. Indeed, we observed that while the difference in gains considering $p$@1 favors CTF-B in more than 6%, this difference in terms of $pavg$@5 is close to 1%.

### 7.8 Combining Strategies

Until now, the studied methods have used only one strategy individually. In order to improve the quality of results, we can also create hybrid methods combining more than one strategy. For example, we can expand the page representation with category descriptors (DESC) and use the category filtering (CTF) over the matched books. The rationale for such a combination comes from the fact that category descriptors contribute to improve recall, whereas category filtering helps to improve precision.

Based on this idea, we implemented methods HYBRID-B, which combines DESC-ALL and CTF-B, and HYBRID-A which combines DESC-ALL-5A and CTF-5A. As shown in Table VI, HYBRID-B is the best method overall, surpassing the methods it combines. This method improves $pavg$@3 and $pavg$@5 in over 3% in comparison with the best single-strategy methods, CTF-B and DESC-ALL. As a result, the gains obtained with this method over the baseline (BOW) are 27% better than the gains obtained with the best single-strategy methods in average precision. With respect to HYBRID-A, we can see in Table VI that the combination of strategies was not so beneficial, as in the case of HYBRID-B.

### 7.9 Taxonomy Impact

The metrics used so far summarize in one single value results obtained for several target pages. To explore a different perspective on the results, we present in Figure 5 the impact of using taxonomies on the expected precision computed for each target page in our collection. In this analysis, we compared our best method HYBRID-B with the baseline method (BOW). The plot is divided into four quadrants, Q1, Q2, Q3 and Q4, in which we can observe: Q1: improvements in many cases where precision were very low or even null; Q2: values of expected precision that were already high and, even so, HYBRID-B presents improvements; Q3: no improvements neither impairments; Q4: few cases in which HYBRID-B impair a good recommendation. In addition, it is noticeable in Q1 and Q2 that the taxonomy-enriched method kept most pages with precision above 0.5.

## 8. CONCLUSIONS

In this work we have shown that the use of human knowledge encoded into a domain-specific taxonomy can significantly improve the quality of content-based recommender systems. We have presented an extended framework for content-based recommender systems that leverages the information encoded into a taxonomy into three strategies. Two of these strategies, the use of category descriptors and classification features, aim at adding new information to the representations of target web pages. Thus, they are expected to favor recall. By its turn, the third strategy, which applies category filter over the recommendation list, is likely to favor precision. Therefore, it totally makes sense to combine these strategies.

We have implemented several recommendation methods that apply those strategies individually and in combination. The methods have been evaluated in a case study where the items to be recommended

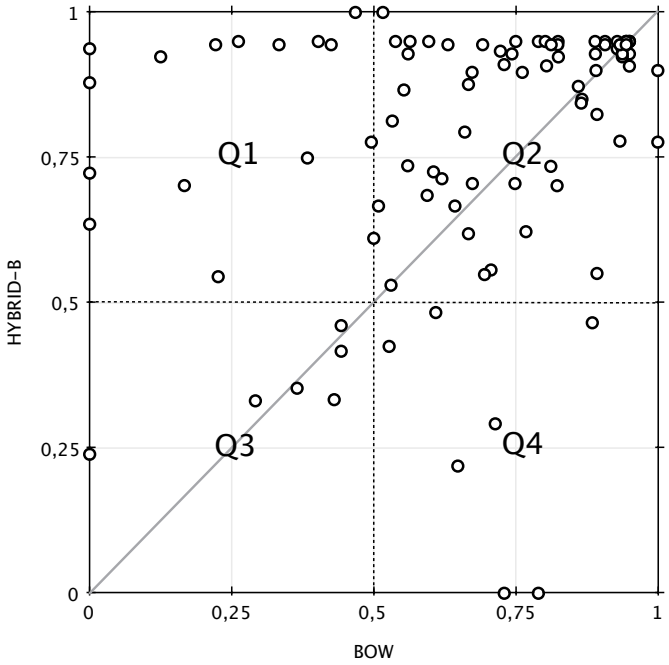


Fig. 5. HIBRID-B versus BOW.

are books and the recommendation target are news pages. We have considered two scenarios: (i) the target news page is automatically assigned torelated categories using a centroid-based classifier; and (ii) the target news page is assigned beforehand to categories from a book taxonomy, e.g., by the author of the news page or by a content editor.

The methods that leverage valuable information encoded in the human-made assignment of the target pages naturally outperform the others. The gain for p@k ranges from 16.13 to 27.87% and the gain for pavg@k ranges from 15.07 to 19.44% both over a baseline based on BOW. The gains were obtained with a confidence level of 99%.

Although in some situations it may be costly to apply manual classification, there are other situations where it is not. For instance, a news web site that also sells books online could have the authors of the news pages assigning categories from a book taxonomy while creating the news pages as well as they do for the related news.

On the other side, the methods that do automatic classification are totally independent of humans. However, automatic classification introduces noise and that is why they do not perform as good as the ones that rely on beforehand classification. The good news is that they also considerably improve the quality of the recommendation list. The gain for p@k goes up to 21.31% and the gain for pavg@k goes up to 13.89%.

In future work, we plan to use machine learning techniques to determine the optimal weights for the recommendation methods considered herein and also to learn different recommendation schemes from the ones proposed in this work. Studying the impact of combining a user profile with the classification features on the recommendation accuracy is also an interesting future work.

## REFERENCES

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

Anagnostopoulos, A., Broder, A., Gabrilovich, E., Josifovski, V., and Riedel, L. Just-in-time contextual advertising. In *Proceedings of the 16th International Conference on Information and Knowledge Engineering*. Lisbon, Portugal, pp. 331–340, 2007.

Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval (Second edition)*. Pearson, 2011.

Buckley, C. and Voorhees, E. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Sheffield, UK, pp. 25–32, 2004.

Carpineto, C., Mori, R., Romano, G., and Bigi, B. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems* 19 (1): 1–27, 2001.

Carpineto, C. and Romano, G. Towards more effective techniques for automatic query expansion. In *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries*. Paris, France, pp. 126–141, 1999.

Carterette, B., Allan, J., and Sitaraman, R. Minimal test collections for retrieval evaluation. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development of Information Retrieval*. Boston, USA, 2006.

Croft, B., Metzler, D., and Strohman, T. *Search Engines: Information Retrieval in Practice*. Pearson Education, 2009.

Gabrilovich, E. and Markovitch, S. Feature generation for text categorization using world knowledge. In *International 11th Joint Conference on Artificial Intelligence*. Edinburgh, UK, pp. 1048–1053, 2005.

Glover, E., Flake, G., Lawrence, S., Kruger, A., Pennock, D., Birmingham, W., and Giles, C. Improving category specific web search by learning query modifications. In *Proceedings of the 1st International Symposium on Applications and the Internet*. Munich, Germany, pp. 23–32, 2001.

Han, E.-H. and Karypis, G. Centroid-based document classification: Analysis and experimental results. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*. Lyon, France, pp. 424–431, 2000.

Hung, L. A personalized recommendation system based on product taxonomy for one-to-one marketing online. *Expert Systems with Applications: An International Journal* 29 (2): 383–392, 2005.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics* 22 (1): 79–86, 1951.

Linden, G., Smith, B., and York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 7 (1): 76–80, 2003.

Oyama, S., Kokubo, T., Ishida, T., Yamada, T., and Kitamura, Y. Keyword spices: a new method for building domain-specific web search engines. In *Proceedings of International Joint Conference on Artificial Intelligence*. Seattle, USA, pp. 1457–1463, 2001.

Pahlevi, S. and Kitagawa, H. Conveying taxonomy context for topic-focused web search. *Journal of the American Society for Information Science and Technology* 56 (2): 173–188, 2005.

Ribeiro-Neto, B., Cristo, M., Golgher, P., and Moura, E. Impedance coupling in content-targeted advertising. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development of Information Retrieval*. Salvador, Brazil, pp. 496–503, 2005.

Rocchio, J. Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System*. Englewood Cliffs, pp. 313–323, 1971.

Salton, G. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989.

Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24 (5): 513–523, 1988.

van Rijsbergen, C. J. *Information Retrieval*. Butterworth, 1979.

Voorhees, E. M. and Harman, D. Overview of the seventh text retrieval conference trec-7. In *Proceedings of the Seventh Text REtrieval Conference*. Gaithersburg, pp. 1–24, 1998.

Yilmaz, E. and Aslam, J. A. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th International Conference on Information and Knowledge Engineering*. Las Vegas, USA, pp. 102–111, 2006.

Ziegler, C., Lausen, G., and Konstan, J. On exploiting classification taxonomies in recommender systems. *AI Communications* 21 (2-3): 97–125, 2008.

Zobel, J. and Moffat, A. Inverted files for text search engines. *ACM Computing Surveys* 38 (2): 1–55, 2006.