Learning to Model Relatedness for News Recommendation

Yuanhua Lv*
Department of Computer Science
University of Illinois at Urbana-Champaign, Urbana, IL 61801
ylv2@uiuc.edu

Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, Yi Chang
Yahoo! Labs, 701 First Avenue, Sunnyvale, CA 94051
{taesup, pranam, zhaohui, xhwang, yichang}@yahoo-inc.com

ABSTRACT

With the explosive growth of online news readership, recommending interesting news articles to users has become extremely important. While existing Web services such as Yahoo! and Digg attract users' *initial* clicks by leveraging various kinds of signals, how to engage such users algorithmically *after* their initial visit is largely under explored. In this paper, we study the problem of post-click news recommendation. Given that a user has perused a current news article, our idea is to automatically identify "related" news articles which the user would like to read afterwards. Specifically, we propose to characterize *relatedness* between news articles across four aspects: relevance, novelty, connection clarity, and transition smoothness. Motivated by this understanding, we define a set of features to capture each of these aspects and put forward a learning approach to model relatedness. In order to quantitatively evaluate our proposed measures and learn a unified relatedness function, we construct a large test collection based on a four-month commercial news corpus with editorial judgments. The experimental results show that the proposed heuristics can indeed capture relatedness, and that the learned unified relatedness function works quite effectively.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering, retrieval models

General Terms

Algorithms

Keywords

Post-click news recommendation, relatedness, learning, connection clarity, transition smoothness, relevance, novelty

1. INTRODUCTION

The recent decade has witnessed an explosive growth of online news. According to a recent report from comScore (http://www.comscore.com/), more than 123 million people visited news websites such as Yahoo! News in May 2010, representing 57 percent of the total U.S. internet audience, with each visitor reading 43 pages on an average. These numbers have been steadily increasing over the past years and show the growing appeal of reading news online.

One important advantage of online news over traditional newspapers is that the former can be augmented with hyperlinks to other related news. When a user has clicked and is reading a current article, he/she may also feel interested in related articles that logically flow from the content of the current page. We thus name it *post-click news recommendation*, with the goal of potentially promoting users' navigation on the visited website. However, on the other hand, if the recommended articles are not logically related to the current news, it would probably not only fail to capture user interest, but also often hurt the user's overall satisfaction. Therefore, the effectiveness of post-click news recommendation is critical to online news websites. While existing Web services such as Yahoo!, Digg, and Google aim to attract users' *initial* clicks by leveraging various kinds of signals [10, 2], how to engage with users algorithmically *after* their initial clicks is largely under-explored.

In the state of the art, post-click news recommendation is typically done by editors' manual effort of searching through a corpus of news documents. The process is not only expensive and cumbersome, but may also be limited by the editors' familiarity with the news topics. Therefore it is highly desirable to develop an effective model or function to deal with the *relatedness* of news articles and find related news *automatically* so as to facilitate post-click news recommendation. However, to the best of our knowledge, no existing method has been proposed to model relatedness directly, other than using a standard information retrieval (IR) model (e.g., [6, 23]) or using news clustering methods (e.g., [35]) to capture *similarity/relevance* instead. In this paper, we study this novel, yet fundamental problem which most online news websites suffer from, and propose effective models to automatically estimate relatedness for recommending related news.

What makes a news article related to the current news article? Presumably, two news articles should be contextually similar to each other. In this sense, *similarity/relevance* would be an important signal of relatedness. This factor can be captured through standard information retrieval models, e.g., [31, 28, 24]. However, similarity/relevance is not a sufficient attribute of the problem at hand, since there are clear differences between news recommendation and traditional information retrieval: for a related article to be recom-

[*] This work was done when the first author was on a summer internship at Yahoo! Labs.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

mended, it would be encouraged that only a fraction of content is shared across the articles, while other parts are mostly about novel news-worthy topics. Nonetheless, IR models usually favor documents that are very similar to the current news (e.g., example 4 in Table 1), yet such documents are not really related to the need of users given that they have already read the current news article. Thus, *novelty*, a somewhat contrary notion to similarity/relevance, should also be a signal of relatedness. Since it is well known that some retrieval functions, e.g., cosine similarity, are also effective measures for novelty and redundancy detection [39, 3], it would be critical to investigate the performance of these measures with "dual-roles" in the relatedness function. Moreover, it is also important to examine if advanced IR models, e.g., passage retrieval [7, 16, 21], which scores a document mainly based on the best matching passage, are more appropriate than standard models in our task.

In addition to the notion of similarity and novelty, we attempt to explicitly capture another key dimension of relatedness, *coherence*, using two novel concepts that we introduce in the following.

First, there should be a *clear connection* between two documents to maintain topical continuity. In other words, the overlap between two documents should be comprised of some meaningful topical context. However, word overlap that is able to capture relevance and novelty may not always capture topical continuity. For the example 5 in Table 1, the overlap of words like "Obama, million, dollar, $\cdots$" are not discriminative enough to represent a specific topic. As a consequence, the two articles are not connected well. To capture this heuristic, we propose a new signal, *connection clarity*, by estimating the entropy of the language usage behind an overlap so as to measure the uncertainty of topics shared by two documents.

Second, the stories in two articles should also be of continuity of senses so that a user's reading interests could *transit* from one to the other smoothly. For example, generally speaking, it would be more smooth to transit from news "Toyota dismisses account of runaway Prius" to "What to do if your car suddenly accelerates" than to "Toyota to build Prius at 3rd Japan plant", although both candidate articles satisfy all the three properties above (i.e., relevance, novelty, and connection clarity). We thus propose the fourth signal *transition smoothness* which measures the likelihood that the two stories can be combined into a single article in a natural way. The intuition is that, if human editors would like to put two stories together under some context, it may mean that the two stories are in a smooth transition, at least from editors' perspective.

We propose four heuristics for characterizing relatedness, namely, relevance, novelty, connection clarity, and transition smoothness. Taking these heuristics as a road map, we design and compare a number of different relatedness measures. We develop a *unified* relatedness function to leverage the complementary relative strengths of various heuristics, by employing a state-of-the-art learning to rank framework, GBRank [40], that combines various relatedness features and learns a unified relatedness function. In doing this, we hope to exploit the strength of machine-learnt rankers, that have demonstrated significant gains over many manually tuned rankers for information retrieval and web search [20].

In order to learn a unified relatedness function and quantitatively evaluate our proposed models, we also develop a larger data set which includes not only a news database, but also judgments of related news for a set of "seed" articles, for the purpose of training and testing. We collect a subset of news articles used by a commercial online news website, design an outline for relatedness judgments, and invite a group of professional news editors to assess the relatedness of the recommended articles. Our analysis shows the good quality of our created data set. The proposed methods are evaluated on this new test collection.

Experimental results show that the four heuristics can characterize relatedness, and that the learned unified relatedness function works quite effectively.

2. PROBLEM FORMULATION

Given the current news that a user is reading, our goal is, informally speaking, to extract from a corpus a list of related articles that the user would like to read afterwards. We next formally define the problem of post-click news recommendation. We begin with a series of useful definitions.

Definition 1 (Seed News) A news article s from the news corpus C that is clicked and being read by a user. It is also called *seed* for conciseness.

We only consider textual news in our context, and usually a news article is comprised of a title, an abstract, and a body. For example, the title of a seed may be "Toyota dismisses account of runaway Prius".

Definition 2 (Candidate News) A candidate news article (or candidate for short) d, is one of news articles from the news corpus C that are recommended to the user after he/she has clicked to read seed s.

For the example above, a reasonable candidate may be the one titled "What to do if your car suddenly accelerates".

Definition 3 (Relatedness Function) The relatedness function f is used to estimate the relatedness score of a candidate d given a seed article s, i.e., $f(s, d)$. It denotes how well the candidate news is related to the seed news. Given that d_1 and d_2 are both candidate news articles, d_1 is a better article to recommend than d_2 if $f(s, d_1) > f(s, d_2)$.

For a recommender system, if the click-rates of different candidate news are known, one could use them directly as the relatedness scores. However, such methods rely heavily on historical data, and thus face a cold-start problem if the seed news is new; moreover, the method would also be biased against selecting a newly-published article as related news if it has not been clicked before. Considering these limitations, in our work, we choose to initially focus on content-based news recommendation, in which recommendations are based on the *relatedness* of news content. However, click-rate as well as other user behavior information could be explored to further improve the relatedness function in future work.

As opposed to relevance feedback in information retrieval [29, 27, 30], which mostly finds similar documents given a feedback document, our relatedness function focuses on a novel notion, "relatedness", which models not only similarity/relevance but also novelty of the candidate story, connection clarity between two articles, and transition smoothness from the seed to the candidate story.

Given a seed article s, the problem of post-click news recommendation is to (1) identify a set of candidate articles $D = \{d_1, \cdots, d_m\}$, and (2) design a relatedness function to score each candidate as $f(s, d_i)$. With D and f, we can then select a ranked list of n ($n < m$) candidates with the highest relatedness scores $D_s = \{d_{s,1}, \cdots, d_{s,n}\}$ for s.

Identifying candidate articles is relatively straightforward. Although similarity/relevance alone is not sufficient to capture relatedness, but it could be a necessary condition, in that, two related documents should at least share some similar content, more or less, to each other. On the other hand, duplicate or near duplicate documents are typically not good candidate documents, so we need to also do a simple redundancy removal to filter them out. We follow the two stage approach proposed in [39] to first use a standard information retrieval model to retrieve a set of documents by taking the seed article as a query, and then remove relatively obvious redundant articles.

However, many challenges need to be addressed in order to design an effective relatedness function. As discussed in Section 1, a related article should satisfy several properties (1) relevance and novelty, (2) connection clarity, and (3) transition smoothness. Relevance and novelty measures have been well studied [28, 33, 24, 39, 3], but it is not clear how well they would work in a unified relatedness function. It is relatively easy to find "connection" between two articles, but it is non-trivial to model the connection clarity without prior knowledge. Furthermore, scoring the transition smoothness from one story to another is also difficult without deep understanding of natural languages; news transition patterns could be useful and may be mined from a set of news transition records collected from a post-click news recommender system, which, however, turns out to be a chicken-and-egg dilemma at the current stage. Finally, even if good measures are designed for all of the above properties, it is still unclear how to develop a unified relatedness function to leverage all these measures effectively; learning to rank [20] may be a possible way, but it needs a large test collection for training and testing of learning to rank algorithms.

3. MODELING RELATEDNESS

Relatedness represents how well a candidate is related to a seed. In order to model relatedness, we propose several signals that characterize relatedness from multiple aspects.

3.1 Relevance and Novelty

Intuitively, a related document should at least share some similar content with the seed document. For example, the two news articles in example 6 of Table 1 are too dissimilar to each other, and clearly they are not related. The notion of relevance in information retrieval, which measures to what extent the topic of a candidate document matches the topic of the query, should be a natural aspect of relatedness. A variety of retrieval models have been well studied in information retrieval to model relevance, such as vector space model, classic probabilistic model, and language models [31, 28, 34, 24, 33, 38]. We will revisit and evaluate some representative retrieval models to examine how well they work for finding *related* articles given a seed article.

On the other hand, if two news articles are too similar to each other, it is clear that users may not be interested in one given that they already read the other, since there may be no novel information in the other article. For example, we do not want to recommend one of the two very similar articles in example 4 of Table 1 as a related news article to the other. Thus, we regard too similar documents as not related in our definition of relatedness and consider novelty as another important signal that is necessary for relatedness. Novelty is often in contrast to relevance, especially in our context where the focus is the relationship between two documents. In fact, many retrieval functions, e.g., cosine similarity, are also effective measures for novelty and redundancy detection [39, 3]. Therefore, we do not distinguish relevance and novelty and use the same set of features for capturing both of them, leaving the machine learning algorithm to balance their "dual-roles".

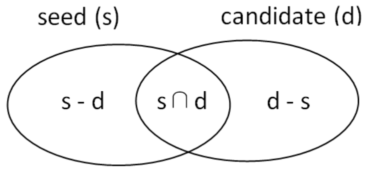

Figure 1: Venn diagram of content overlap between two documents

We evaluate four standard retrieval functions: cosine similarity [31], BM25 [28], language models with Dirichlet prior smoothing [24, 38], and language models with Jelinek-Mercer smoothing [24, 38]. They cover the most popular information retrieval models. Among all, BM25 and language models with Dirichlet prior smoothing represent two state-of-the-art models in information retrieval [11]. Cosine similarity, one of the earliest retrieval functions, does not work as well as the other three ones. However, we select it instead of other more advanced variations of the vector space model, e.g. [34], mainly because it has also been shown to be one of the most effective measures for novelty detection [39]. We choose language models with Jelinek-Mercer smoothing, as it has worked very well for verbose queries [38]; in our task, the seed document (query) is also usually long and verbose.

All the above standard retrieval functions capture both relevance and novelty in a single "relevance" score. As a result, if we use such a "relevance" score to rank documents, without properly balancing their two roles (i.e., relevance and novelty) well, the top-ranked documents may be dominated by redundant or unrelated articles, which would hurt the precision at top documents.

Passage retrieval [7, 16, 21] can relax this problem to some degree. For a related document, it would be encouraged that only a portion shares similar content with the seed article, while other parts are mostly about some *novel* topics. We thus investigate passage retrieval, which computes the relevance score for a document mainly based on the best matching passage. Intuitively a score based on passage retrieval would not be influenced so much as traditional document-based retrieval by the existence of novel information. Therefore, we hypothesize that passage retrieval would work better than standard retrieval models in terms of precision at top-ranked documents.

3.2 Connection Clarity

Both relevance and novelty signals can only model the word overlap between two articles s and d; they essentially measure the size of overlap, i.e. $|s \cap d|$, as shown in Figure 1. However, it is often observed that pure word overlap between two news articles does not necessarily indicate *relatedness.* We illustrate this with the following example:

s: *White House: Obamas earn* $5.5 *million in 2009*

d: *Obama's oil spill bill seeks* $118 *million, oil company*

The two articles s and d share the same words like "Obama, million, dollar, ...", and there is also novel information contained in the candidate d, but they turn out to be unrelated.

Intuitively, in order to make two documents related, they should share a clear story/thread to make them topically cohesive. This can often be achieved by repeating the same topic or similar topic, which forms topical links that connect two documents together and make them related. However, *word* overlap may not necessarily form a clear *topic* overlap. For the above example, a set of words like "Obama, million, dollar, ..." are not discriminative enough to represent a specific topic. We thus propose another signal, *connection clarity*, to capture this heuristic. Arguably the clearer the topical connection is, the more related the two documents would probably be.

One straightforward way for modeling connection clarity is to consider the topic distributions of two articles. Specifically, we can first identify topics associated with each document using probabilistic topic models, such as PLSA [14] and LDA [5], and then calculate the negative KL-divergence between two distributions as the connection clarity score:

$$\textit{clarity(s, d)} = -\sum_{t=1}^{N} P(t|s) \log \frac{P(t|s)}{P_{\lambda}(t|d)} \quad (1)$$

where t denotes a topic and N is the number of total topics (in our work, we use LDA [5] and set $N = 1000$ empirically). $P_{\lambda}(t|d) = (1-\lambda)P(t|d) + \lambda P(t|C)$ is a linearly smoothed topic distribution associated with document d, where the topic distribution of the whole collection $P(t|C)$ is selected as the background model. Although there is no zero-probability problem, we use a smoothing strategy to decrease the effects of non-discriminative topics (i.e., topics associated with many documents) [38]. Intuitively, if both articles share the same clear/discriminative topic, the score will be high. However, this method suffers from a drawback: it essentially estimates the connection *size* rather than connection *clarity*; as a result, two documents sharing many uncorrelated or loosely correlated topics, though no topic has a large probability, may also receive high scores, which is, however, counterintuitive.

Thus we seek a more reasonable method that can measure the connection clarity directly based on the language usage of the "connection" (i.e., $s \cap d$ in Figure 1). We propose to use the *entropy* of the language model (unigram distribution over words) behind $s \cap d$ to measure the *uncertainty* of topics shared by two documents. Intuitively, if $s \cap d$ is about a clear and discriminative story, the language model is usually characterized by large probabilities for a small number of topical terms, while if $s \cap d$ does not imply a clear connection (e.g., only consisting of uncorrelated words), the language model would be smoother. So, the entropy will be smaller in the former case than the latter.

We first define $s \cap d$, the overlap of s and d, in a more formal way. Since $s \cap d$ is represented as a bag of words here, we can use counts of words to define it. Let $c(w, d_i)$ be the count of word w in document d_i. Then $c(w, s \cap d)$ is

$$c(w, s \cap d) = \min\{c(w, s), c(w, d)\} \quad (2)$$

It is often not very accurate to estimate a language model $P(w|s \cap d)$ based on the text of $s \cap d$ alone, since $s \cap d$ may only contain a small set of words. We thus borrow the idea of the relevance model proposed by Lavrenko and Croft [18] to estimate $P(w|s \cap d)$ using more evidences. The relevance model has been shown to perform very well for estimating an accurate query language model in information retrieval [22]. The relevance model $P(w|s \cap d)$ is estimated in terms of the joint probability of observing any term w together with terms from $s \cap d$. With an assumption that w and all words from $s \cap d$ are sampled independently and identically to each other, the relevance model is computed as follows:

$$P(w|s{\cap}d) \propto P(w, s{\cap}d) \propto \sum_{r \in R} P(w|r) \prod_{w' \in V} P(w'|r)^{c(w', s\cap d)} \quad (3)$$

where r is a document or the model estimated from the corresponding single document, R is the set of documents that are associated with $s \cap d$, V is the vocabulary of the whole news collection, and $\prod_{w' \in V} P(w'|r)^{c(w, s\cap d)}$ is the likelihood of an individual document language model generating $s \cap d$.

Note that the idea of the relevance model is to use the likelihood of generating $s \cap d$ of an individual document as the weight, and estimate a final language model based on weighted aggregation of term counts in the document set R. To improve both efficiency and effectiveness, we restrict R to only contain top 50 documents that are most likely to generate $s \cap d$.

We next compute the entropy of $P(w|s \cap d)$. Since the entropy of $P(w|s \cap d)$ is often affected significantly by common terms (e.g., 'the', 'and', ...), we compute a relative entropy or KL-divergence, similarly to [9]. The relative entropy as the connection clarity score, essentially measures the language usage associated with $s \cap d$ as compared to the background language model of the collection as a whole.

$$\textit{clarity(s, d)} = \sum_{w \in V} P(w|s \cap d) \log \frac{P(w|s \cap d)}{P(w|C)} \quad (4)$$

Based on this formula, a larger score of connection clarity indicates higher relatedness between two documents.

3.3 Transition Smoothness

Even if a candidate article satisfies all the above properties, would a user like to transit to this article after reading the current news? Let's take a look at some examples:

s *: Toyota dismisses account of runaway Prius*

d*₁: What to do if your car suddenly accelerates*

d*₂: Toyota to build Prius at 3rd Japan plant: report*

Given a seed (s), we believe that users would be more likely to read candidate 1 (d_1) than candidate 2 (d_2). One possible reason is that d_1 and s are more coherent, and we can even imagine that it would be natural to combine s and d_1 into one coherent article. Although, d_2 also connects well to s via "Toyota Prius", it would be odd if these two stories are combined, because there is little *continuity of senses* between stories; as a result, a user's reading interests would less likely to transit from s to d_2 smoothly.

Thus we propose another signal, *transition smoothness*, to model relatedness from another dimension. Transition smoothness measures how well a user's reading interests can transit from s to d. As shown in Figure 1, since $s \cap d$ is shared by two documents, intuitively we are more interested in the transition smoothness from $s - d$ to $d - s$, i.e., from the "known" information to the "novel" information. We define $s - d$ and $d - s$ as follows:

$$c(w, s - d) = c(w, s) - c(w, s \cap d) \quad (5)$$

$$c(w, d - s) = c(w, d) - c(w, s \cap d) \quad (6)$$

It is nontrivial to model transition smoothness directly, which needs deep understanding of natural languages. Instead, we attempt to estimate it in an alternative way. Imagine that if, in some context, an editor would like to put two "stories" (not necessarily natural language stories, since they are just bags of words) $s-d$ to $d-s$ into one news article, it probably means that the two stories have a smooth transition, at least from that editor's perspective. Therefore, through analyzing a large collection of news articles, if two stories often occur in similar contexts, it may suggest that they are smooth in transition. And now the problem is how we can model and compare "contexts" for $s-d$ to $d-s$.

Our first approach is to model context using associated documents. Such a document-based context is represented as a vector of documents. Specifically, the context of $s-d$, i.e., $\vec{C}_{s-d}$, is a $|D|$-dimension vector, where $|D|$ is the total count of articles. The weight of each dimension d_i is defined as $g(s-d, d_i)$. g can be any positive function. In our work, we use the relevance score of document d_i with respect to "query" $s-d$ as $g(s-d, d_i)$. In particular, we use the well-known BM25 retrieval function [28] to compute relevance scores. This implies that if a document is more relevant to $s-d$, the document would play a more important role to determine the context of $s-d$. Similarly we can estimate the context vector $\vec{C}_{d-s}$ for $d-s$. Finally, the transition smoothness score can be computed using the cosine similarity between context vectors:

$$smoothness(s,\ d) = \frac{\vec{C}_{s-d} \cdot \vec{C}_{d-s}}{||\vec{C}_{s-d}|| \cdot ||\vec{C}_{d-s}||} \tag{7}$$

An alternative approach is to model a word-based context. The word-based context is represented as a unigram language model. Specifically, the context of $s-d$ and $d-s$, i.e., $P(w|s-d)$ and $P(w|d-s)$, can also be estimated using the relevance model [18] in a similar way as we estimate $P(w|s \cap d)$ in Section 3.2. Then we can compare two contexts in terms of their divergence. Due to the zero-probability problem, we use the Jensen-Shannon divergence [19]:

$$\begin{aligned} smoothness(s,\ d) = &\frac{1}{2}\sum_{w \in V} P(w|s-d) \log \frac{P(w|s-d)}{P(w|M)} \\ &+ \frac{1}{2}\sum_{w \in V} P(w|d-s) \log \frac{P(w|d-s)}{P(w|M)} \end{aligned} \tag{8}$$

where $P(w|M) = 0.5 \cdot P(w|s-d) + 0.5 \cdot P(w|d-s)$.

We can see that both methods boil down to computing the similarity/distance between the contexts of two documents. The main difference is that the first method uses a document-based context representation, while the second uses a word-based representation.

4. CONSTRUCTING A UNIFIED RELATEDNESS FUNCTION

In order to learn a unified relatedness function and quantitatively evaluate our proposed models, we need a data set which includes not only a news database and a large set of seed news articles, but also judgments of related news for each seed article for the purpose of training and testing. There is an existing small collection [6], which, however, only contains 50 seed articles with no quality analysis of their judgments. We thus decide to establish a larger test collection and collect editorial judgments which reflects a more realistic scenario. Furthermore, through constructing a test collection, we also seek if our definition of relatedness can be effectively identified by human assessors and to what degree different assessors agree with each other in making judgments.

Next, we describe how we constructed our editorial data set and learn our relatedness function in detail.

4.1 Constructing a Test Collection

We collected a subset of Yahoo! News articles from March 1st to June 30th 2010. Then we randomly generated a list of 549 seed news from June 10th to June 20th, 2010 with at least $2,000$ visits. We chose the seed articles from mid-June period since a news article often has a very short active life and thus the recommended news in reality could be mostly comprised of older news (i.e., news occurring before mid-June) and sometimes a few newer ones in the near future (i.e., late-June). The restriction of avoiding less popular news as seed articles was to reduce the cases that some assessors might be unfamiliar with the news stories so as to feel uncertain to make judgments; in real applications, we may also expect our models to be most useful for such relatively popular news. So our data collection strategy reflects the real world applications well.

To generate relatedness judgments for each seed article efficiently, we used the pooling technique [15] and asked assessors to judge only a pool of top-ranked documents returned by various runs (i.e., recommender systems). We used several standard information retrieval models as the "runs", including cosine similarity with a raw TF and a log IDF [31], Okapi BM25 ($k1 = 1.2$, $k3 = 1000$, $b = 0.5$) [28], language models with Dirichlet prior smoothing ($\mu = 2,000$) [24, 38], and language models with Jelinek-Mercer smoothing ($\lambda = 0.9$) [24, 38], to retrieve documents from the collection. The retrieval parameters were shown to work well on TREC7 ad hoc test collection (http://trec.nist.gov/), also a news corpus, for which we used the verbose "narrative" portion of TREC7 topics as queries. We mainly used the body portion of both candidate and seed articles for indexing, leading to 4 runs. In order to improve diversity, for the language modeling approach with Dirichlet prior smoothing method, we also added two other runs where the title and abstract (provided by Yahoo! News) portions of seed articles were indexed as "queries" respectively. Finally, we generated 6 runs in total.

Next, we did a simple redundancy detection using the two-stage approach proposed in [39] to filter out relatively obvious redundancy. Specifically, we used cosine similarity, which had been shown to work effectively in [39], to remove near duplicate documents of which the similarity score passes a pre-specified redundancy threshold. In our work, we set the redundancy threshold to 0.8 empirically.

One difference of our task from traditional information retrieval tasks is that we are only interested in the top-ranked documents. Therefore, for each seed article, the top-ranked 15 documents (after redundancy filtering) from each run were pooled together to be judged, resulting in a set of $16,484$ news documents in total to be judged for 549 seed news. Moreover, to investigate the consistency of assessor judgments, we randomly selected 2 documents for each seed, which would be judged by two different assessors, while all other documents would be judged by only one assessor.

Very related	Somewhat related	Unrelated	Redundant
3695 (21.0%)	4277 (24.3%)	9278 (52.8%)	322 (1.8%)

Table 2: The distribution of relatedness judgments.

4.1.1 Guidelines for Editorial Judgments

We invited a group of *professional news editors from a commercial online news website* to participate in the judging tasks. Though they are all experts of news articles, we still trained them carefully to understand the tasks of post-click news recommendation. After that, they were asked to judge the relationship of a candidate article to its corresponding seed news as one of the 4 degrees: very related, somewhat related, unrelated, and redundant. The meaning and description of each degree are introduced below, and some examples are also shown in Table 1.

Very Related: *The two documents should share some interesting topic/event threads. The candidate should contain somehow complementary information to the seed be it narrower or broader in scope. Imagine that the user thinks up some questions while reading the seed article, and the recommended article provides a good answer to (at least) one of them. Very related content is serendipity at its best.*

Somewhat Related: *The candidate contains is at least related to the topic of the seed, but not very interesting. After reading the seed article, the user would not be particularly well-informed or eager to read the candidate article. On the other hand, if we choose to provide the candidate, the user would understand why we displayed it, rather than thinking "how on earth is that related?"*

Unrelated: *There is no clear connection between the candidate and the seed. The candidate looks like a spam.*

Redundant:[1] *The candidate discusses the same thing as the seed without providing any clearly novel information.*

4.1.2 Analysis of Editorial Judgments

We received 17,572 valid judgments in total, and a few documents which made editors hard to judge were removed from the set. The number and proportion of judgments from each relatedness level are reported in Table 2. We can see that "unrelated" documents take the most portion, followed by "somewhat related" and "very related" documents. After a simple redundancy removal step, redundancy seems not to be a serious concern in our task, since it only takes 1.8%.

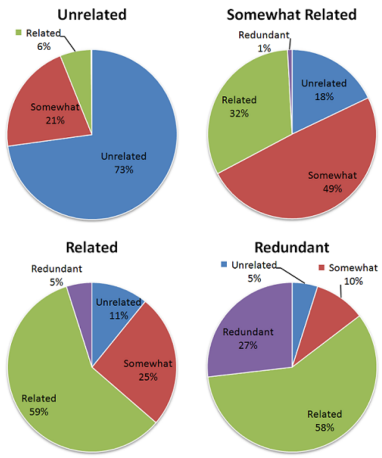

Figure 2: The distribution of relatedness judgments given by one editor, when another one's judgment is "unrelated" (top left), "somewhat related" (top right), "very related" (bottom left), and "redundant" (bottom right), respectively.

We also had 1098 documents, each of which was judged by 2 different editors. We show some consistency analysis in Figure 2. It is observed that different editors are more consistent in identifying unrelated and very related documents, while they often do not agree with each other too much on redundant and somewhat related documents, especially on redundant documents. Except the "redundant" level, the multi-level judgments indeed provide more information than binary judgments. For example, we are more confident to take a document as related document if an editor labels it as "very related" than that the editor labels it as "somewhat related". Overall, the agreement between editors is 62.5% for our four-level relatedness judgments. However, if we only consider binary judgments by simply taking very related, somewhat related, and redundant judgments all as "related", then the agreement is increased to 79.9%, which is as high as the agreement of binary relevance judgments by TREC assessors [13]. It suggests that our editorial judgments are of a good quality.

We argue that the inconsistency of some judgments may be mainly due to the different "criteria" of editors to make multi-level judgments. To investigate it, we further look into the agreement of *relative* relatedness of two documents. Specifically, we first ignore seed articles of which either of the two candidate documents is judged as "redundant" by any editor, so as to make the relative ranking easier to generate. After that, we have 494 out of 549 seed articles left. We then assume that two documents can be ranked randomly if they are judged to be equally related (i.e. tie), otherwise "very related" $\succ$ "somewhat related" $\succ$ "unrelated". Finally we find that the agreement ratio is 80.8%, much higher than that of absolute judgments. The high agreement in relative relatedness inspires us to learn relatedness functions from the pair-wise preference information implied in judgments.

We use a 4 point relatedness scale, where "very related", "somewhat related", "redundant", and "unrelated" judgments receive ratings 3, 2, 1, and 0, respectively. Such a rating strategy for relatedness assessments is chosen based on both our intuition and the probability ranking principle [26]: "very related" articles should be ranked the highest, "unrelated" should be ranked the lowest, a "redundant" document is still better than an "unrelated" one that may hurt user experience, and a "redundant" document should be worse than a "somewhat related" document that contains novel information. For any document with two different judgments, we select a judgment with the higher agreement ratio according to Figire 2 as its final judgment. For example, if a document is judgment as "very related" and "somewhat related" by two editors respectively, we then select "very related" as its final judgment. Yet this rating scale may not be optimal, but it has been shown to work very well empirically.

4.2 Learning a Relatedness Function

Given the features described in Section 3 and editorial data obtained as in Section 4.1, we use a state-of-the-art

[1] Although we have attempted to filter out "duplicate"/"near duplicate" documents, there still could be documents that are redundant in content but very different in presentation.

Relatedness	Seed news	Candidate news	
Very related	Polish leader, 96 others dead in Russia jet crash	List of some of those who died in the crash of the Polish presidential plane in Russia	1
	Toyota dismisses account of runaway Prius	What to do if your car suddenly accelerates	2
Somewhat related	Toyota dismisses account of runaway Prius	Toyota to build Prius at 3rd Japan plant: report	3
Redundant	Obama, McCain pledge to work together for reform	Obama, McCain vow to work together, reform government	4
Unrelated	White House: Obamas earn $5.5 million in 2009	Obama's oil spill bill seeks $118 million, oil company tax increase	5
	Polish leader, 96 others dead in Russia jet crash	Tax rows roll on in election campaign	6

Table 1: Examples for relatedness judgments, where the title of each news article is shown.

learning to rank algorithm, namely GBRank [40], to leverage features and develop a unified relatedness function.

More specifically, denote $\mathbf{x}_i \in \mathbb{R}^d$ as the feature vector for the i-th document pair (i.e., a seed and a candidate), r_i as the editorial label of relatedness, and $f : \mathbb{R}^d \to \mathbb{R}$ as a relatedness function. Then, for total n editorial data, we define a loss function $L(\{\mathbf{x}_i\}, \{r_i\}, f)$ as

$$\begin{aligned} L(\{\mathbf{x}_i\}, \{r_i\}, f) = &\sum_{(i,j)\in\mathcal{P}_n} \Big((r_i - r_j) - (f(\mathbf{x}_i) - f(\mathbf{x}_j))\Big)_+^2 \\ &+ \lambda \sum_{(i,j)\in\mathcal{T}_n} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 \end{aligned} \qquad (9)$$

where $(x)_+$ is x if $x \geq 0$ and 0 otherwise, $\mathcal{P}_n$ is the set of preference pairs, and $\mathcal{T}_n$ is the set of tied pairs. We then use the stochastic functional gradient boosting technique [12] to minimize Equation 9 in the functional space using regression trees as weak learners and obtain the relatedness function. The used parameters for functional gradient boosting trees are number of nodes $N = 10, 15$ for tree weak learners, number of trees $T = 600$, shrinkage parameter $\nu = 0.03, 0.05, 0.07$, data sampling rate $s = 0.5, 0.7, 0.8$, and tied pairs weight $\lambda = 0.5, 1, 3$.

The reason why we chose the preference based, pair-wise algorithm is from the analysis in Section 4.1.2 that our editorial labels tend to be consistent more in the relative sense.

5. EXPERIMENTS

We preprocess both candidate and seed documents by applying the Porter stemmer and removing stopwords using a total of 418 InQuery stopwords. The average length of news bodies is not very long, only 97 words after preprocessing. We use different methods to rank those candidate documents that have been judged by editors. We choose the widely-used NDCG as our main measure and report NDCG scores at top 1, 3, 5, and 10 documents, since we are especially interested in the performance at top documents.

5.1 Comparing Individual Retrieval Models

To compare different retrieval models, we use the body text to index candidate documents (i.e., body indexing), which has been shown to work well [6]. For seeds (queries), we test title, abstract, and body indexing respectively.

We first report the comparison of different retrieval models in Table 3, where the parameters of different models are tuned to work well on the collection. We can see that, among different methods to index seed documents (queries), body indexing tends to be better than abstract indexing which is better than title indexing. This observation is consistent to the findings of [6] using a small data set with binary judgments. It suggests that the title and abstract may lose useful information since they are "compressed".

Comparing different retrieval models, BM25 is clearly the best when using title or abstract indexing for seed documents, but when using body indexing, the two language modeling approaches work as well as BM25. Besides, cosine similarity performs surprisingly as well as or even better than language models in some cases, but its NDCG1 score appears relatively poor. This may be due to the effectiveness of cosine similarity for redundancy detection [39], which often brings redundant documents to the top position.

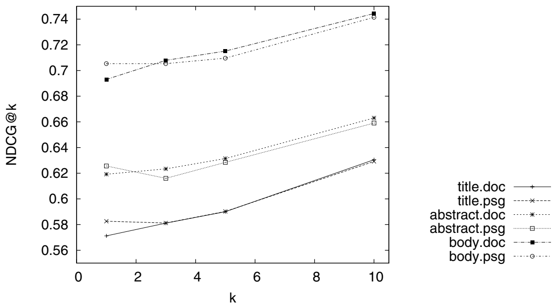

Figure 3: Comparison of document (doc) and passage (psg) retrieval. In a label "X.Y", 'X' stands for indexing methods for seed documents.

We next evaluate passage retrieval to see if it is more appropriate in our task. We use a strategy of fixed-length arbitrary passage retrieval (passages that can start at any position in a document but with the fixed length) [16], which has been shown to work effectively [16, 21]. We fix the passage length to 250 empirically. Following [21], we also use the language modeling approach with Jelinek-Mercer smoothing method for passage retrieval. Experiment results are summarized in Figure 3. It shows that document-based retrieval is better in many cases, probably because our news documents are mostly homogeneous and short so that passage retrieval does not help [21]. However, it is interesting to see that passage retrieval improves NDCG1 clearly. It may indicate that passage retrieval somehow relaxes the concern of ranking redundant documents on top. Probably because we only have 1.8% redundant documents after a simple redundancy removal step, the benefit of passage retrieval does not improve NDCG3, 5, and 10.

5.2 Comparing Machine-Learned Relatedness Models

In the following experiments, we evaluate relatedness models which are learned using GBRank [40] to leverage a set of features. We use a 5-fold cross validation strategy, where the total 549 seed documents (queries) are split into 5 collections randomly. And the performance scores reported below are based on the combination of 5 testing collections.

We first combine standard retrieval models (as listed in Table 3) using GBRank, and the model obtained is labeled

Metric	Title				Abstract				Body			
	lm-dir	lm-jm	cosine	BM25	lm-dir	lm-jm	cosine	BM25	lm-dir	lm-jm	cosine	BM25
NDCG1	0.5859	0.5712	0.5660	0.6093	0.6253	0.6192	0.6076	0.6340	0.7057	0.6930	0.6841	0.7027
NDCG3	0.5783	0.5812	0.5836	0.6050	0.6171	0.6234	0.6104	0.6334	0.7033	0.7079	0.6887	0.7042
NDCG5	0.5911	0.5900	0.5958	0.6162	0.6319	0.6314	0.6269	0.6440	0.7098	0.7152	0.6983	0.7133
NDCG10	0.6285	0.6305	0.6369	0.6536	0.6615	0.6630	0.6624	0.6753	0.7421	0.7443	0.7330	0.7448

Table 3: NDCG comparison of standard information retrieval models with different portions of seed documents as "queries". "lm-dir" and "lm-jm" stand for language models with Dirichlet prior smoothing and Jelinek-Mercer smoothing respectively.

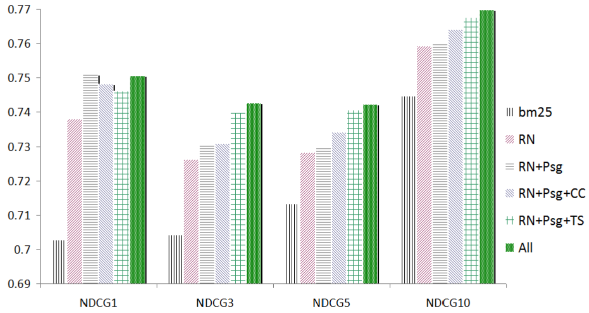

Figure 4: Performance comparison of machine-learned recommenders using different feature sets.

Metric	BM25	RN	All
NDCG1	0.7027	0.7380^{+}	0.7507^{+*}
NDCG3	0.7042	0.7262^{+}	0.7426^{+*}
NDCG5	0.7133	0.7282^{+}	0.7422^{+*}
NDCG10	0.7448	0.7593^{+}	0.7698^{+*}

Table 4: Performance comparison of different relatedness functions. '+' and '*' mean the improvements of model All over both BM25 and RN are statistically significant respectively.

	Feature name	Imp.	method	s	d
1	cosine.body	100	cosine	body	body
2	lm-jm.body	70.75	lm-jm	body	body
3	BM25.body	45.10	BM25	body	body
4	smooth-1.body	36.59	Formula 7	body	
5	psg-abs	35.78	passage	abstract	body
6	BM25.title	31.54	BM25	title	body
7	clarity.title	29.32	Formula 4	title	
8	smooth-2.abs	25.59	Formula 8	abstract	
9	BM25.abs	22.49	BM25	abstract	body
10	smooth-2.title	21.53	Formula 8	title	
11	lm-dir.body	21.07	lm-dir	body	body
12	clarity.body	20.43	Formula 4	body	
13	psg.title	19.15	passage	title	body
14	clarity.abs	19.07	Formula 4	abstract	
15	smooth-2.body	18.04	Formula 8	body	

Table 5: Top-15 important features. "Imp." means the importance score, and the last two columns indicate the indexing methods for seeds and candidates.

as "RN". The comparison of RN with BM25 is shown in Figure 4. We see that RN improves consistently and significantly over BM25, and the improvement is especially large for NDCG1. Since NDCG1 is most sensitive to the existence of redundant documents, it may suggest that the learned model RN can balance relevance and novelty well.

We then train another model by adding passage retrieval features in. This model is labeled as "RN+Psg". We can see from Figure 4 that, although RN+Psg does not introduce clear improvements to RN on NDCG5 and NDCG10, it indeed boosts NDCG1 significantly. This observation confirms our previous finding that passage retrieval handles redundant documents well.

Next, we introduce another heuristic, connection clarity. We investigate the proposed two instantiations of this heuristic, i.e., Formula 1 and 4, respectively. Our experiments show that the first instantiation (Formula 1) does not help and even often hurts NDCG1 and NDCG3, probably due to its drawback as we have discussed in Section 3.2. We then drop the first instantiation and add the second one, which is labeled as "RN+Psg+CC". We find that the second approach is indeed better than the first one. From Figure 4, we can see that RN+Psg+CC improves NDCG10 and NDCG5 over RN+Psg, though NDCG1 is decreased slightly. It may suggest that connection clarity somehow tends to favor redundant documents, since the "connection" between duplicate stories is just this story, which would often be "clear".

We now turn to evaluate the last heuristic, transition smoothness. There are also two instantiations, document-based context comparison and word-based context comparison. We add them into the feature set together, since they essentially boil down to computing transition smoothness in a similar way. The new model is named "RN+Psg+TS". We can see from Figure 4 that RN+Psg+TS outperforms RN+Psg clearly in most cases. Yet similar to RN+Psg+CC, this model also fails to improve NDCG1.

Comparing the two novel heuristics, transition smoothness seems to work better than connection clarity. Anyway, there is much room to improve both of them through further optimizing their feature implementations.

Finally, we combine all heuristics and construct a *unified relatedness function*, labeled as "All". The performance is shown in both Figure 4 and Table 4. Clearly, this unified model performs the best among all methods. We can see that the improvement of "All" over "RN" is close to the improvement of "RN" over the most effective single method BM25. And the improvements are also statistically significant using the Wilcoxon test ($p < 0.05$). It indicates that our heuristics indeed capture more evidences of "relatedness" by going beyond relevance and novelty.

5.3 Analyzing the Unified Relatedness Model

We go a step further to analyze the importance of different heuristics and features in contributing to the model. Our unified relatedness model is learned using GBRank, which is an extension of the gradient boosting decision tree [12]. We thus follow the relative importance measure proposed in [12] to examine the proposed features. The top-15 influential features and their corresponding importance scores are reported in Table 5.

It shows that all the proposed heuristics contribute clearly to the model. Among all, relevance and novelty together

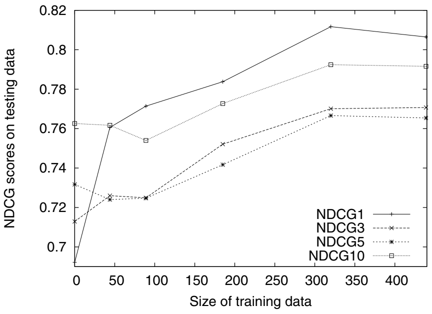

Figure 5: Sensitivity to the size of training data.

tend to be most important. It is interesting to see that, though cosine similarity is significantly worse than BM25 as an individual relatedness function, the former turns out to be the most important feature in the unified model. One possible explanation is that cosine similarity is also very effective in redundancy and novelty detection [39]. Language models with Jelinek-Mercer smoothing method (lm-jm) also seems to be more important than BM25, which needs more experiments to further examine the reason. Although not the most important one, BM25 contributes 3 important features, probably due to its excellent performance for modeling relevance. Besides, passage retrieval also appears important.

Moreover, it is very encouraging to see that the proposed two novel heuristics, connection clarity and transition smoothness, contribute 7/15 important features together. Regarding connection clarity, it seems that the connection of titles between two articles is more useful, which looks interesting because intuitively users also tend to judge related articles mainly based on titles. For transition smoothness, document-based context comparison (smooth-1, we only implemented one feature based on smooth-1) tends to be better than word-based ones (smooth-2), which suggests that a document may be a more meaningful and appropriate unit for capturing contexts than words. Overall, transition smoothness looks more important than connection clarity, which is consistent with our previous observation of their empirical performance.

Since we use a supervised learning approach, one routine question would be: how much training data is needed to train a reliable model? To examine it, we draw in Figure 5 the NDCG curves for the number of training instances for one of the 5 folds, where the total number of training instances is 439. It is interesting to see a general upward trend as we use more training data (zero training data means that only BM25 is used). And we can often learn a reliable relatedness model with more than 200 training instances. However, there seems to be a plateau after the size of training data reaches 300. Yet further experiments and larger test collections are needed to confirm this observation.

As for real application, efficiency is very important. However, there is no serious efficiency concern with our work. First, the number of news articles generated per day is not very large. Secondly, we only need to score and re-rank a few candidate documents for a seed article. Thirdly, feature scores can also be pre-computed, and post-click news recommendation can also be done offline. As the news collection evolves over time, some features of the proposed two novel heuristics, i.e., connection clarity and transition smoothness, may be updated periodically. However, it still does not hurt the efficiency too much; since the life of a news story is usually very short, we only need to update features for a few "active" articles.

6. RELATED WORK

Recommendation has been studied extensively in the past. Content-based filtering and collaborative filtering are the two main types of recommendation techniques that have seen most interest [1]. In news domain, recommendation services often rely on user activities such as click patterns to find the most popular news [2], user-news ratings for collaborative news recommendation [25, 17, 10], or user interests for content-based and adaptive news access [4]. All these techniques focus on attracting users' *initial* clicks and visits, but how to engage with users *after* their initial click has not been well explored.

Due to the characteristics of news, such as short life and rich content, content-based recommendation is presumably a good choice for post-click news recommendation, at least early in the life-time of a news article when there is lack of related news preference data. However, traditional content-based filtering, e.g., [4], is usually to recommend stories which reflect users' long-term interests well (e.g., a user might generally like sports articles), while our work focuses on recommending news related to users' ad hoc interests implied by the currently clicked news, although the learned relatedness model would also benefit traditional content-based filtering.

There are existing studies that also attempt to recommend news articles or blog posts given a seed article [6, 36, 23]. However, their methods mainly use standard IR models [6, 23] or other similarity-based functions [36] to score candidate news. Prior work has not studied how to characterize relatedness. A recent work [32] focused on connecting two news stories in two different time points through finding a set of coherent stories in the middle. Their objective function is maximizing the coherence of the whole story chain, while we maximize the relatedness of recommended articles to a seed article. We believe our work of pair-wise relatedness would benefit their tasks, in that, any neighboring nodes in their chain should be related to each other.

The process of post-click news recommendation appears similar to traditional information retrieval [28, 33, 24]. However, as opposed to IR, which mostly finds relevant documents given a query, our relatedness function focuses on a novel notion, "relatedness", which models not only similarity/relevance but also novelty of the candidate story, connection clarity between two articles, and transition smoothness from the seed to the candidate story.

In comparison with traditional IR tasks, novelty detection is highly related to our work. Existing studies in novelty detection, e.g., [8, 39, 37, 3], often aim to locate relevant and novel information through finding a tradeoff between relevance and novelty. Most work boils down to balancing relevance and novelty in an ad-hoc way, such as the two-state novelty detection approach proposed in [39]. The notion of "relatedness" studied in our work, however, depends on multiple signals beyond just relevance and novelty. Thus, it is difficult to tune their tradeoff and adjust their interactions using similar methods. We thus define a set of signals and features which can capture multiple aspects of relatedness and use a learning approach to leverage all these features to model relatedness and rank candidate articles.

Recently, learning to rank [40, 20] has attracted much attention in IR. The advantage of machine learning is that it provides an optimization framework to leverage various kinds of signals. Learning to rank has been shown to be quite effective in IR and Web search [20] to model relevance of documents. In contrast to arguably all work on learning to rank, our study investigates the capability of learning approach to model a novel notion, relatedness.

7. CONCLUSIONS

In this paper, we studied post-click news recommendation. More specifically, given a user has read a current news article, our idea is to automatically identify "related" news articles which the user would like to read afterwards. We focused on characterizing and modeling the notion *relatedness*. A large test collection was created with editorial judgments of relatedness, multiple heuristics from various dimensions were proposed to measure relatedness, and a unified relatedness function was learned. The analysis shows the good quality of the constructed collection; experimental results demonstrate that the proposed heuristics can indeed capture relatedness from multiple perspectives, and that the learned unified relatedness function can be able to leverage relative strengths of various relatedness measures and works quite effectively for news recommendation.

This research is only a first attempt at post-click news recommendation, so there are many open problems for future research. Although our content-based relatedness measures worked well in our experiments, we believe that the underlying relatedness relationship is document and user specific, and that document and user adaptive measures will eventually be more accurate. It is also very promising to incorporate into the unified relatedness function other non-content features, such as click-rate, user comments, social networks, news transition patterns, timestamp, and document source, which will be important sources of evidence for further improving relatedness modeling.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.

[2] D. Agarwal, B.-C. Chen, and P. Elango. Explore/exploit schemes for web content optimization. In *ICDM '09*, pages 1–10, 2009.

[3] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR '03*, pages 314–321, 2003.

[4] D. Billsus and M. J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[6] T. Bogers and A. van den Bosch. Comparing and evaluating information retrieval algorithms for news recommendation. In *RecSys '07*, pages 141–144, 2007.

[7] J. P. Callan. Passage-level evidence in document retrieval. In *SIGIR '94*, pages 302–310, Dublin, Ireland, 1994.

[8] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, pages 335–336, 1998.

[9] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR '02*, pages 299–306, 2002.

[10] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW '07*, pages 271–280, 2007.

[11] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *SIGIR '04*, pages 49–56, 2004.

[12] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[13] D. Harman. Overview of the third text retrieval conference (trec-3). In *TREC*, 1994.

[14] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99*, pages 50–57, 1999.

[15] K. S. Jones and C. J. van Rijsbergen. Report on the need for and the provision of an 'ideal' information retrieval test collection. Technical Report (British Library Research and Development Report No. 5266), Computer Laboratory, University of Cambridge, 1975.

[16] M. Kaszkiel and J. Zobel. Effective ranking with arbitrary passages. *Journal of the American Society for Information Science and Technology*, 52(4):344–364, 2001.

[17] K. Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference*, 1995.

[18] V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR '01*, pages 120–127, 2001.

[19] J. Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Infor. Theory*, 37:145–151, 1991.

[20] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[21] X. Liu and W. B. Croft. Passage retrieval based on language models. In *CIKM '02*, pages 375–382, McLean, Virginia, USA, 2002.

[22] Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of CIKM '09*, 2009.

[23] C. Macdonald, I. Ounis, and I. Soboroff. Overview of trec-2009 blog track. In *TREC '09*, 2009.

[24] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, 1998.

[25] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94*, pages 175–186, 1994.

[26] S. E. Robertson. The probability ranking principle in ir. pages 281–286, 1997.

[27] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society of Information Science*, 27(3):129–146, 1976.

[28] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC '94*, pages 109–126, 1994.

[29] J. J. Rocchio. Relevance feedback in information retrieval. In *In The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall Inc., 1971.

[30] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society of Information Science*, 41(4):288–297, 1990.

[31] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[32] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *KDD '10*, pages 623–632, 2010.

[33] A. Singhal. Modern information retrieval: a brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24:2001, 2001.

[34] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96*, pages 21–29, 1996.

[35] H. Toda and R. Kataoka. A clustering method for news articles retrieval system. In *WWW '05*, pages 988–989, 2005.

[36] Y. Yang, N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias. Query by document. In *WSDM '09*, pages 34–43, 2009.

[37] C. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03*, pages 10–17, 2003.

[38] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.

[39] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *SIGIR '02*, pages 81–88, 2002.

[40] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for web search. In *NIPS '07*. 2007.