To Each His Own: Personalized Content Selection based on Text Comprehensibility

Chenhao Tan[†], Evgeniy Gabrilovich[‡], Bo Pang[‡]
[†] Cornell University, Ithaca, NY 14853, USA
[‡] Yahoo! Research, 4301 Great America Parkway, Santa Clara, CA 95054, USA
chenhao@cs.cornell.edu | {gabr | bopang}@yahoo-inc.com

ABSTRACT

Imagine a physician and a patient doing a search on antibiotic resistance. Or a chess amateur and a grandmaster conducting a search on Alekhine's Defence. Although the topic is the same, arguably the two users in each case will satisfy their information needs with very different texts. Yet today search engines mostly adopt the one-size-fits-all solution, where personalization is restricted to topical preference. We found that users do not uniformly prefer simple texts, and that the text comprehensibility level should match the user's level of preparedness. Consequently, we propose to model the comprehensibility of texts as well as the users' reading proficiency in order to better explain how different users choose content for further exploration. We also model topic-specific reading proficiency, which allows us to better explain why a physician might choose to read sophisticated medical articles yet simple descriptions of SLR cameras. We explore different ways to build user profiles, and use collaborative filtering techniques to overcome data sparsity. We conducted experiments on large-scale datasets from a major Web search engine and a community question answering forum. Our findings confirm that explicitly modeling text comprehensibility can significantly improve content ranking (search results or answers, respectively).

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models and Selection Process

General Terms

Algorithms, Experiments

Keywords

Personalization, text comprehensibility, user modeling, re-ranking

1. INTRODUCTION

Many factors explain users' choices in content consumption, while observable human behavior often produces a single judgment of utility (e.g., a click) that conflates different factors. Most prior studies focused on modeling users' topical preferences (either long-term or short-term). Such models, however, may not be able to explain why users prefer some items over others within the same topic. For instance, an aeronautics professor and an aspiring technology journalist writing about rocket science will likely choose to read very different texts.

We conjectured that users often choose content—consciously or unknowingly—based on its *comprehensibility*. The notion of comprehensibility is inherently user-specific, as reading more involved texts may require users to possess adequate background. It is, of course, also topic-specific; for instance, a renowned professor might also be a beginner cook, and would therefore prefer easy-to-understand cooking instructions. In this paper, we build models of text comprehensibility as well as of users' reading proficiency, and use users' past preferences to predict future content choices. It is exactly the need to disentangle users' topical and comprehensibility preferences that makes this problem difficult.

Given a text fragment, our aim is to measure its degree of sophistication, that is, how easy (or difficult) it is to read for the majority of users. To this end, we build a classifier for predicting text comprehensibility, which uses a host of text readability features motivated by research in computational linguistics (see Section 2.1). A note on terminology is in order here. The word "readability" is used in the literature in a number of different senses, ranging from the degree of grammaticality of the text, to the degree of fluency in (automatically-constructed) search summaries [19], to the degree of difficulty of text as judged by average sentence length and vocabulary size [21, 11, 24, 15]. Here, we use "readability" in the latter sense, but to avoid possible confusion we opted to use the term "comprehensibility" instead.

We model users' reading preferences by analyzing the content they chose to read in the past. One way to do so is to compute the average comprehensibility of the texts read by each user. A crucial limitation of this approach, however, is that the absolute comprehensibility scores produced by the classifier are not necessarily comparable across texts on different topics. To circumvent this limitation, we only compare comprehensibility scores of related texts (e.g., those returned in response to the same search query), and use such pairwise judgments to compute the probability that the user prefers easier texts. As one would expect, our experiments show that the improvement in personalized content ranking is larger when the user's comprehensibility preference is more pronounced (i.e., when the above probability is substantially different from 0.5).

Modeling each user's overall preference allowed us to observe that not everyone prefers easy texts (cf. Section 6.1). However, this general model is necessarily too coarse, hence we refine it by computing topic-specific comprehensibility preferences. Doing so, however, is difficult, since the more specific the model is, the more data is needed to reliably compute it, while many users might have

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.

only read a few texts in each topic, or none at all in some topics. We tackle this data sparsity by using collaborative filtering techniques.

We use the output of the comprehensibility classifier for improving the ranking of content in two different scenarios, namely, Web search and community question answering (CQA). In both cases, we use historical data (click logs for Web search, and best answer tags for CQA forums), and develop models that better explain past user behavior. It should be noted that using historical data at test time puts us at an inherent disadvantage owing to the position bias, because a highly-ranked item might have been clicked merely due to its prominent position rather than true utility. Nonetheless, using our comprehensibility classifier leads to significant ranking improvements on both datasets, despite this disadvantage.

We observe the largest ranking improvements for informational queries [5] in topics such as "Hobbies & Interests", "Arts & Entertainment", and "Real Estate". Predictably, our approach does not offer much improvement for transactional queries in topics such as "Retail", where other factors (such as product price or brand recognition) may dominate users' utility judgments.

Modeling the level of text sophistication is an under-studied area, and we proposed a principled approach to personalized content selection via modeling text comprehensibility. The contributions of this paper are threefold. We developed a comprehensibility classifier for predicting the sophistication level of a given text, and used it to build topic-specific models of users' reading proficiency. Our approach is entirely based on implicit user feedback, and we evaluated a number of strategies for extracting users' comprehensibility preferences from their historically-observed actions. Finally, our experiments on two large-scale datasets confirm the utility of quantifying text comprehensibility for improved content ranking.

2. METHODOLOGY

Our goal is to better satisfy non-topical aspects of users' information needs in a variety of content selection tasks, ranging from personalized ranking of Web search results to selecting the best answer (for the asker) in community question-answering forums. In this section, we use the Web search setting as our running example.

Here we focus on one particular type of preference in content selection, namely, comprehensibility of the relevant content. Different users may have their individual preferences for easier or more sophisticated text, and our aim is to augment relevance-based ranking with comprehensibility-based personalization. To this end, we need to be able to (a) rank texts that are related to the same narrow topic (e.g., search results for a query) by their comprehensibility level, (b) model whether a given user prefers an easier or more sophisticated content for her current information need, and (c) produce a personalized ranking. In the remainder of this section, we describe the algorithms used to address each of these tasks.

2.1 Estimating text comprehensibility

Comprehensibility classifier The simplest way to perform comprehensibility-based ranking for a set of documents is to build a classifier that assigns a comprehensibility score to any input document. Texts that are topically related to an information need can then be ranked by their comprehensibility scores to produce a comprehensibility-based ranking.

It is possible that a global, "topic-agnostic" classifier may not fully capture topic-dependent aspects of reading difficulty. For simplicity, we introduce a light-weight classifier based on data easily available online; more sophisticated topic-specific classifiers can be considered in future work.

We are not aware of existing large-scale labeled resources that are publicly available and cover a broad range of topics. Thus, we construct a labeled dataset by extracting pages from Simple English Wikipedia[1] (denoted as W_S) and English Wikipedia[2] (denoted as W_{en}). Articles in Simple Wikipedia are written using *Basic English*[3], a subset of English with a restricted vocabulary and simple grammar rules. Many articles in W_S can be aligned to corresponding articles in W_{en} with the same title. Upon discarding overly short articles, we find 40,032 aligned article pairs[4]. We label articles from W_S as *easy*, and those from W_{en} as *hard*, and then train a logistic regression classifier.

Table 1: Readability index features used in the classifier.

Flesch [21]:	$206.835 - 1.015\frac{\#\text{words}}{\#\text{sentences}} - 84.6\frac{\#\text{syllables}}{\#\text{words}}$
Flesch-Kincaid [23]:	$0.39\frac{\#\text{words}}{\#\text{sentences}} + 11.8\frac{\#\text{syllables}}{\#\text{words}} - 15.59$
Gunning FOG [15]:	$0.4(\frac{\#\text{words}}{\#\text{sentences}} + 100\frac{\#\text{polysyllables}}{\#\text{words}})$
ARI [21]:	$4.71\frac{\#\text{characters}}{\#\text{words}} + 0.5\frac{\#\text{words}}{\#\text{sentences}} - 21.43$
SMOG [24]:	$1.043\sqrt{30\frac{\#\text{polysyllables}}{\#\text{sentences}}} + 3.1291$
Coleman-Liau [11]:	$0.0588L - 0.296S - 15.8$ (L: #letters per 100 words; S: #sentences per 100 words)

Polysyllables are words with 3 or more syllables.

We use as features several standard readability indices [21, 11, 24, 23, 15] (Table 1), where word length and sentence length are used as coarse proxies of semantic difficulty and syntactic complexity. We also include the bag of words features. In order to build a classifier that is applicable to a broad range of texts, we limit our vocabulary to the "Basic English 850 word list"[5]. These unigram features are weighted by term frequency and L_2-normalized.

Comprehensibility score (S_c) The output of the classifier is the likelihood of the article being hard to read, which we refer to as the *comprehensibility score* (S_c).

We report two kinds of performance metrics. First, we compute standard classification accuracy by comparing S_c with the (global) threshold of 0.5, which amounts to a binary decision on whether the text is "easy" or "hard". However, scores computed for very different topics are not necessarily comparable, hence we also report the accuracy of pairwise score comparison for each title, that is, whether the article from W_S received a lower score than the corresponding one from W_{en}. The following table reports the above metrics using 5-fold cross-validation.[6]

	Global threshold	Per-title comparison
Accuracy	88.3%	97.4%

As we can see, while the accuracy of using a global threshold is quite high, the accuracy of pairwise comparison is even higher, as

[1] http://simple.wikipedia.org
[2] http://en.wikipedia.org
[3] http://simple.wikipedia.org/wiki/Basic_English
[4] When extracting textual content from HTML, we discard segments with fewer than 100 characters, and then discard documents with fewer than 50 words.
[5] http://simple.wikipedia.org/wiki/Wikipedia:Basic_English_ordered_wordlist
[6] We also conducted a small-scale evaluation of S_c scores in the Web search scenario, where one of the authors judged the relative difficulty of a pair of search results for 20 queries. 4 of the 20 document pairs were found difficult to judge; on the remaining cases the agreement between the annotator and the classifier was 81.25%.

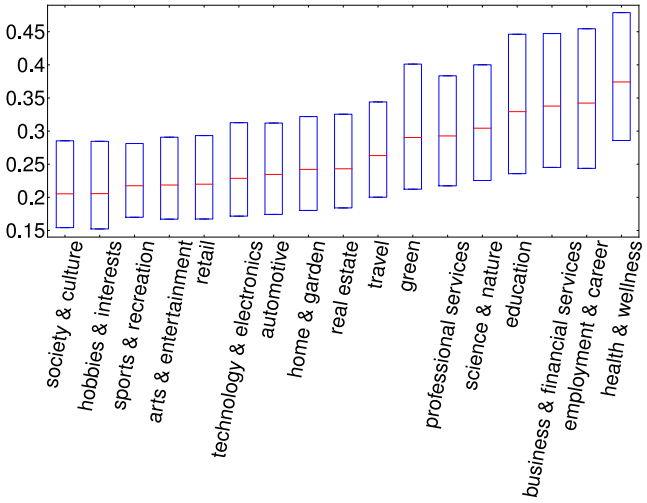


Figure 1: Comprehensibility score (S_c) distribution for different topics. Blue bars indicate 25% and 75% percentiles; red lines indicate 50% percentile (median).

we get the correct "order" over 97% of the time. This indicates that S_c is more reliable for comparing comprehensibility of texts on the same narrow topic than for texts on different topics. To verify this point, we examine the topical variance of S_c in more details.

Topical variance of S_c Figure 1 shows the topical distribution of S_c scores for 154 million Web pages (see Section 3.1 for more details on this dataset). As we can see, a relatively easy article in the "health & wellness" category might receive a higher S_c score than a relatively hard article in "hobbies & interests". This is consistent with our previous observation that it is more meaningful to use S_c for comparing documents within the same topic.

2.2 Comprehensibility-based content selection

Intuitively, different users may have different comprehensibility preferences, which can further vary on a per-topic basis. We capture such preferences by building user profiles, which are then used to modify the relevance-based ranking of available content. We now describe these two steps in turn.

2.2.1 Modeling user comprehensibility preferences

We describe three methods for building user profiles. We begin with a coarse model that captures a user's general (topic-independent) comprehensibility preference, that is, whether the user tends to prefer easier or more sophisticated content among search results returned for their queries.

Topic-independent profile (basic model)

For a user u, let $a >_u b$ denote the user's preference of content item a over b. Suppose we have a method (which we describe in Section 2.3) to obtain a set of n preference pairs for each user:

$$\Omega_u^{pref} = \{(< a_i, b_i >, w_i) \mid a_i >_u b_i, \text{with weight } w_i\}.$$

We start with the simple case with equal weights ($w_i = 1$) for all pairs. Consider a random variable X with a Bernoulli distribution parameterized by p, which takes value 1 if the user prefers the harder content. Assume we have a sample of size n ($X_1 = x_1, ..., X_n = x_n$), corresponding to n preference pairs of content items, where

$$x_i = \begin{cases} 1, & S_c(a_i) > S_c(b_i) \\ 0 & S_c(a_i) < S_c(b_i) \end{cases}$$

(remember the pairs are ordered to reflect the user's preference for a_i over b_i, i.e., $a_i >_u b_i$). We now estimate p, the probability of user u preferring harder content (we denote this estimator as P_u). Let $k = \sum_i x_i$. Since for most users we are dealing with relatively small n, the maximum likelihood estimator (k/n) is undesirable. Instead, we take the Laplace estimator of p, which uses Uniform$(0, 1)$ as the prior distribution. The posterior estimation of P_u is thus a function over Ω_u^{pref}:

$$P_u = f(\Omega_u^{pref}) = \frac{k+1}{n+2}$$

Again, in the simplest case, we select content for each user according to her estimated P_u, regardless of the topic. We denote this topic-independent user model as BASIC.

To incorporate weights over the pairs, we can compute a weighted version:

$$P_u^{(w)} = f^{(w)}(\Omega_u^{pref}) = \frac{k^{(w)}+1}{n^{(w)}+2},$$

where $k^{(w)} = \sum_i w_i x_i$, $n^{(w)} = \sum_i w_i$. Clearly this reduces to P_u when $\forall i, w_i = 1$. Each of the models described in the remainder of this section can be similarly modified to have a weighted and an unweighted version.

Topic-dependent profile

Suppose a set of candidate documents (e.g., top results for a search query) can be classified into an existing topic hierarchy, where the node "default" is the root. For each user, we construct a set of pairwise preferences for each topic $\Omega_{u,t}^{pref}$ from Ω_u^{pref}.

We define an order relationship between two topic nodes in this hierarchy as follows:

$$t_2 <_h t_1 \Leftrightarrow t_2 \text{ is a descendant of } t_1$$

For each preference pair $pp_i \in \Omega_u^{pref}$, let t_i be its topic (e.g., the topic of the query associated with the pair), and let

$$\Omega_{u,t}^{pref} = \{pp_i \in \Omega_u^{pref} | t_i \leq_h t\}.$$

For any reasonably sized $\Omega_{u,t}^{pref}$ (i.e., $|\Omega_{u,t}^{pref}| > \theta$), we compute $P_{u,t} = f(\Omega_{u,t}^{pref})$ similarly to the BASIC model above.

Clearly, we will not obtain enough observations for every possible (u, t) pair to reliably compute $P_{u,t}$, especially for the topics residing deep down in the hierarchy. As a coarse approximation, we can fall back to using the topic-independent (BASIC) model if necessary. For an incoming query, we consider the top-level topic node t corresponding to the topic predicted for this query. If $P_{u,t}$ can be estimated, we use $P_{u,t}$ for personalized content selection; otherwise, we fall back to P_u. We refer to this model as TOPICAL.

Collaborative Filtering

In order to alleviate this data sparseness problem (i.e., lack of topic-specific preferences) in a more principled manner, we build upon the work in collaborative filtering. That is, if there exist certain correlations between the comprehensibility preferences for some topics, then we can analyze the observed comprehensibility preferences over all (user, topic) pairs, and predict comprehensibility preferences for unseen ones.

Formally, let n_u be the number of users, and n_t be the number of topics. We construct a matrix $G^{n_u \times n_t}$, where G_{ij} is the likelihood of user i preferring harder content in topic j as estimated from observed data. Note that for cells (i, j) with no observed data, $G_{i,j}$

should receive the same value as an "average" user. We achieve this by computing the global mean of all $P_{u,t}$ values (estimated from observed data) as $g = \frac{1}{\sum_u \sum_t I(P_{u,t} \neq 0)} \sum_u \sum_t P_{u,t}$, and let

$$G_{ut} = \begin{cases} P_{u,t} - g, & \Omega_{u,t}^{pref} \neq \emptyset \\ 0, & \text{otherwise.} \end{cases}$$

This way, the cells with no observations and the cells corresponding to the "average" user receive the value of 0.

Note that this formulation is slightly different from the standard collaborative filtering setting, where the non-zero entries in the matrix are actual observations (e.g., ratings given by a user to an item). In our case, we manipulate estimates ($P_{u,t}$) rather than actually observed values, and the estimates are for a small number of topics rather actual observations over a much larger set of "items", as in the standard setting.

We adopt the maximum-margin matrix factorization approach from the collaborative filtering literature, where we compute an approximation of G with a low-rank decomposition $U^T V$. That is, we want to minimize the following objective function:

$$\sum_{i,j,G_{ij} \neq 0} ||U^T V_{(ij)} - G_{ij}||^2 + ||U||_F + ||V||_F$$

We use CofiRank [36] to solve this optimization problem. Once we obtain the optimal U and V, we compute

$$G^{cf} = U^T V + g.$$

For test queries, we first proceed as in TOPICAL, but when $P_{u,t}$ cannot be reliably estimated, we fall back to G_{ut}^{cf} computed using the above collaborative filtering approach. The default P_u is only used when the topic of the content can not be determined (e.g., a query receiving the "default" category). We denote this user model as COLLABORATIVE.

2.2.2 Combining the Rankings

We cast the content selection problem as a ranking problem. In the Web search scenario, we have an initial relevance-based ranking (as returned by the search engine), and we want to adjust it to better satisfy users' comprehensibility preferences. In the community question-answering scenario, our task is to rank answers so that the answer chosen as the best one by the asker ranks at the top. In the latter case, there might not exist any native ranking of answers on the site, and we can either produce a ranking purely based on comprehensibility preferences or implement quality-based measures to produce an initial ranking.

Here, we consider the general case where we have an original topic-relevance-based ranking R over a set of documents D. Previous work in the personalized search literature has combined topical relevance with (topical) personal preferences by simply computing the sum of the global relevance score (R) and the personalized ranking score. Our approach differs in several aspects.

Let $R(d)$ be the rank of $d \in D$ given by R. Let R_u be the ranking over D in descending order of the comprehensibility score S_c. Here, we slightly abuse the notation and use P_u to denote a user profile built using any of the three models introduced in Section 2.2.1. We then produce the combined ranking by ordering items in ascending order of the following value

$$R(d) + \beta * (2 * P_u - 1) * R_u(d) \quad (1)$$

First, we have a global parameter β, which controls the relative importance of comprehensibility (R_u). We also vary the importance of R_u depending on how pronounced the user's comprehensibility preference is, that is, how much P_u deviates from 0.5. Furthermore, we need to reverse R_u if the user prefers easier content ($P_u < 0.5$). Thus, we multiply the second term by $(2 * P_u - 1)$ to incorporate personalized adjustment over the global β.

2.3 Generating pairwise preferences

We now describe corpus-dependent methods for generating pairwise comprehensibility preferences for two different types of datasets, Web search click logs and best answers in CQA sites.

2.3.1 Web search click log

As discussed in Section 7, click logs are often used for extracting user preferences. Using different models of how users browse search result pages, prior work has examined different ways of reconstructing user preferences that account for position bias [17, 18, 27, 10]. We considered three methods for extracting pairwise preferences: two as proposed by Joachims et al. [18], and a variation over the second method (which is new to the best of our knowledge).

Let's consider a toy example to examine the differences between the three methods. Suppose we have a search result page with 5 results ($l_1, l_2, ..., l_5$). Suppose the user clicked on l_2 and l_4.

The first method is the classical Click > Skip above. Here the assumption is that users browse results in the order of presentation, and each clicked result is "better" than those presented earlier (i.e., viewed) and not clicked. For our toy example, this yields three preference pairs: $l_2 >_u l_1$, $l_4 >_u l_1$, and $l_4 >_u l_3$.

DEFINITION 1. *Click > Skip above (*CSA*)*
For a ranked list $(l_1, l_2, l_3, \ldots)$ and clicked position set $\mathcal{C}$ for user u, we define

$$l_j >_u l_i, \quad \textbf{if } i < j, i \notin \mathcal{C} \text{ and } j \in \mathcal{C}.$$

The second method accounts for the noise in click data, and only trusts the last clicked item to be "better" than those skipped. For our toy example, this yields $l_4 >_u l_1$ and $l_4 >_u l_3$.

DEFINITION 2. *Last click > Skip above (*LCSA*)*
For a ranked list $(l_1, l_2, l_3, \ldots)$, a clicked position set $\mathcal{C}$ for user u, and the position of the (temporally) last clicked item LC, we define

$$l_j >_u l_i, \quad \textbf{if } i < j, i \notin \mathcal{C} \text{ and } j = LC.$$

The third method makes a stronger assumption that the last item the user clicked is the one that finally satisfied the user's information need, while all the items above it, including those clicked, are inferior. For our toy example: $l_4 >_u l_1$, $l_4 >_u l_2$, and $l_4 >_u l_3$.

DEFINITION 3. *Last click > All above (*LCAA*)*
For a ranked list $(l_1, l_2, l_3, \ldots)$, a clicked position set $\mathcal{C}$ for user u, and the position of the last temporally clicked item LC, we define

$$l_j >_u l_i, \quad \textbf{if } i < j \text{ and } j = LC.$$

Preference weighting Traditionally, when pairwise preferences are used in learning to rank, each pair carries the same weight. Intuitively, the closer in position two results are, the more likely they are similar in their topical relevance, and the more confidence we have that the implied preference is due to non-topical (i.e., comprehensibility) reasons. Thus, for each pairwise preference $l_j >_u l_i$, we compute a weight as a function of their distance $w = 2^{-(j-i-1)}$. For example, using LCSA, we would obtain $(l_4, l_1, 0.25)$ and $(l_4, l_3, 1)$.

2.3.2 Best answers in CQA forums

Consider a community-based Q&A site such as Yahoo! Answers, where the asker can label one of the answers as the best answer. If there are n answers for a question, we can simply form a preference pair between the best answer and all the other $n - 1$ answers.

Since n can vary greatly from question to question, and we want the asker's preference on each question to carry roughly the same weight, we take each preference pair with weight $1/n$.

3. DATA

3.1 Search dataset

Our search dataset was sampled from one month (May, 2011) of Yahoo! Web Search query logs. After filtering adult content, the dataset underwent the following pre-processing steps.

In order to focus on search intents that are meaningful for comprehensibility-based personalized search, we filtered out navigational queries using an automatic navigational query classifier, as well as all search sessions that only resulted in one click on the first result, which are also highly likely to have navigational intents.

We then crawled the content of all Web pages returned as one of the top 10 results for queries in our data[7], and applied our comprehensibility classifier to these pages. It should be noted that the user's decision to click on a search result is based on the snippet presented on the search result page, before they view the content of the chosen URL. However, applying the comprehensibility classifier to the search snippets is problematic since most snippets are broken text segments, and those that happen to have longer "sentences" (in order to include search terms) will have higher S_c. Instead, we aggregated all Web pages at the domain level, and used the domain-averaged S_c as the smoothed S_c for each page. Intuitively, this captures the comprehensibility reputation of a domain, which can influence users' clicking decisions. Indeed, previous work has found aggregation of information at the domain level to be beneficial [25, 7].

Each page in our dataset was classified into a proprietary class hierarchy, which had 17 top-level nodes (cf. Figure 1), and 216 nodes in total; "default" was assigned to pages when the classifier had low confidence. The statistics shown in Figure 1 were computed over the 154,650,334 pages with non-default class assignments. Similarly, a class label from the same topic hierarchy was assigned to each query based on its search results [6].

The processed data was then split into three parts chronologically: the first 20 days were used as training data, the next 5 days as development data, and the last 5 days as test data.

Clearly, we cannot expect to produce reasonable personalization for users who have barely clicked on any results before. Thus, we limited our study to users with at least 10 queries and at least one click logged in the training data. We randomly selected 424,566 users to form our search dataset.

3.2 Answers dataset

Yahoo! Answers is a community question-answering site. An *asker* posts a question, which may receive multiple answers from other users. The asker has an option of choosing one of the answers as the best answer.[8] We focused on questions with more than one answer, among which a best answer was chosen by the asker. We obtained a dump of Yahoo! Answers data between January, 2010 and April, 2011, where all questions (and answers) in 2010 were used as the training data, and the 4 months in 2011 were used as the test data (development data was not needed here, as there are no parameters to tune for this dataset). We randomly selected 85,172 users who have posted at least 10 questions in the training data, and restricted the test set to these users. Our dataset consists of a total of 4.9 million questions and 39.5 million answers. Each answer in the dataset received an S_c score from the comprehensibility classifier.

[7]Queries with fewer than 8 results were removed from the dataset.

[8]If the asker does not pick the best answer, a community vote is opened to choose the best answer. We disregarded such cases.

Table 2: 18 configurations: 3 methods to generate pairwise preferences × weighted / unweighted × 3 user profile models

Preference generation methods		Weighting	User profile models
CSA	(Click > Skip above)	Weighted	BASIC
LCSA	(Last click > Skip above)	Unweighted	TOPICAL
LCAA	(Last click > All above)		COLLABORATIVE

4. EXPERIMENTS ON SEARCH DATA

We now evaluate the effectiveness of our approach in Web search.

4.1 Evaluation Measures

Following Dou et al. [14], we employ two measures to evaluate our approach to personalized search.

Average Clicked Rank In Web search, average clicked rank for query s is defined as

$$AvgRank_s = \frac{1}{|C_s|} \sum_{p \in C_s} R(p),$$

where C_s denotes the set of clicked Web pages for query s, and $R(p)$ denotes the rank of page p. The final average rank for the entire test query set S is the average of $AvgRank_s$ values:

$$AvgRank = \frac{1}{|S|} \sum_{s \in S} AvgRank_s$$

Lower *average clicked rank* values indicate better performance.

Rank Scoring The score of a ranked list of Web pages is defined as

$$R_s = \sum_j \frac{\delta(s,j)}{2^{(j-1)/(\alpha-1)}},$$

where j is the rank of a page in the list, $\delta(s,j)$ is 1 if page j is clicked for the test query s and 0 otherwise, and α is set to 5 [14, 31, 4]. The final rank scoring is aggregated over all test queries as follows:

$$R = 100 \frac{\sum_s R_s}{\sum_s R_s^{Max}}$$

R_s^{Max} is the maximum possible score when all clicked pages appear at the top of the ranked list. **Larger** *rank scoring* value indicates better performance.

4.2 Experimental setup

We report performance for the following systems:

- Baseline: The original ranking in which the search engine presented results to the users. Only the top 10 results from the original ranking were passed to our methods to compute personalized re-ranking.
- Our approach: We presented three different ways to generate pairwise preferences from click logs (Section 2.3.1), which can be either weighted or unweighted, and three user profile models (Section 2.2.1). In total, we have $3 * 2 * 3 = 18$ configurations of our methods (summarized in Table 2). The threshold θ (described in Section 2.2.1) was set to 5.

Under the evaluation measures defined in Section 4.1, our method is at a disadvantage compared to the baseline owing to the position bias: if we re-rank a result that was previously displayed at position 10 (with no clicks observed) to a position higher in the

page, this penalizes our performance, but it could be that the user would have liked this result had they viewed it. One might want to restrict the re-ranking to only position 1 through the lowest clicked position. This avoids penalizing us unfairly by excluding results that might not have been viewed by the user; however, this would also give us an unfair advantage: a random re-ranking within this set is very likely to improve over the baseline (according to the above metrics). Consequently, in our experiments we report the former comparison, which is to our disadvantage: re-ranking top 10 results and comparing against the original ranking using the metrics described in Section 4.1.

User saliency For ease of presentation, we also define the notion of user preference *saliency* as

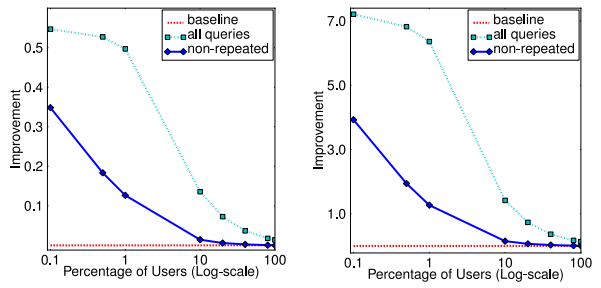
$$Q_u = |P_u - 0.5|,$$

where for each given configuration, P_u is the user preference computed under that particular combination of parameters. We expect that for users with higher saliency, the improvement of comprehensibility-based personalization would also be more pronounced, and we would be interested in the improvements obtained in each bucket of user saliency. To this end, for each configuration of our method, we rank users according to Q_u, and report improvement over the baseline for top $k\%$ users, for different values of k.[9]

We tuned the β parameter (Eq. 1, Section 2.2.2) on the development set, and obtained the best performance with $\beta = 0.4$. Thus, the importance of comprehensibility-based ranking is roughly between 20% and 30% compared to the original topical-relevance-based ranking, for users with extremely pronounced preferences.

4.3 Results

First, we take one of the configurations (WEIGHTED-LCAA +COLLABORATIVE) to make a few high-level observations. Figure 2 plots the improvement measured in both average clicked rank and ranking score. We observe that the improvement is indeed more pronounced for users with higher saliency. Computed over all queries for the top 0.1% most salient users, the average improvement is more than 0.5 in averaged clicked rank, and close to 7.0 in ranking score. While the improvement is smaller for less salient users, we do obtain a significant improvement for all users,



(a) Average Clicked Rank (b) Ranking Score

Figure 2: Improvement over baseline: computed on all queries and on non-repeated queries only.

[9]Note that since the top $k\%$ users are different for different configurations of our method, the corresponding baseline for that subset of the users can also be different. Instead of comparing raw performance numbers across different methods, we only report the improvement over the corresponding baseline performance, which is more comparable across different methods.

Note, however, that a significant portion of queries submitted by users are *repeated* queries [14, 33]. Previous work has shown that using a memory-based method can be highly effective for repeated queries [14]. While our method is not directly "remembering" the actual URLs clicked by the user in the past for repeated queries, it can indirectly promote the previously clicked URLs.[10] Thus, performance improvement over repeated queries may not reflect the utility of comprehensibility preferences, and for the rest of Section 4, we focus our analysis on **non-repeated** queries.

As shown in Figure 2, while the improvement over non-repeated queries is less pronounced, we do obtain a similar trend, which is consistent in both measures. As discussed in Section 4.2, we focus on the improvement over the baseline. To provide a reference point, the performance of the baseline system is 3.6 using average clicked rank, and 74.4 using ranking score for the entire test set.

Next, we examine the improvements achieved by different configurations (on non-repeated queries) in more details.

4.3.1 *Overall analysis over non-repeated queries*

Table 3 shows paired t-test results (using average clicked rank) for different configurations of our method versus the baseline, for top 10% and 50% (ranked by Q_u), and all users.

First, for the top 10% most salient users, all configurations significantly outperform the baseline. All improvements are statistically significant; many are highly significant with $P < 0.001$. Even when we consider the entire set of users, several variations yield highly significant improvements. That is, modeling user preferences in text comprehensibility does help achieve better content ranking in Web search, despite the adversarial experimental setup as explained in Section 4.2.

The effect of weighting the preference pairs also turns out to be quite prominent. Visually, we can readily notice there are more stars in the WEIGHTED section of the table than in the UNWEIGHTED section. Compared over 50% of all users, the UNWEIGHTED versions tend to be less effective in beating the baseline than their WEIGHTED counterparts. This conforms to our intuition that preferences based on search results that are closer in position are more useful in modeling comprehensibility.

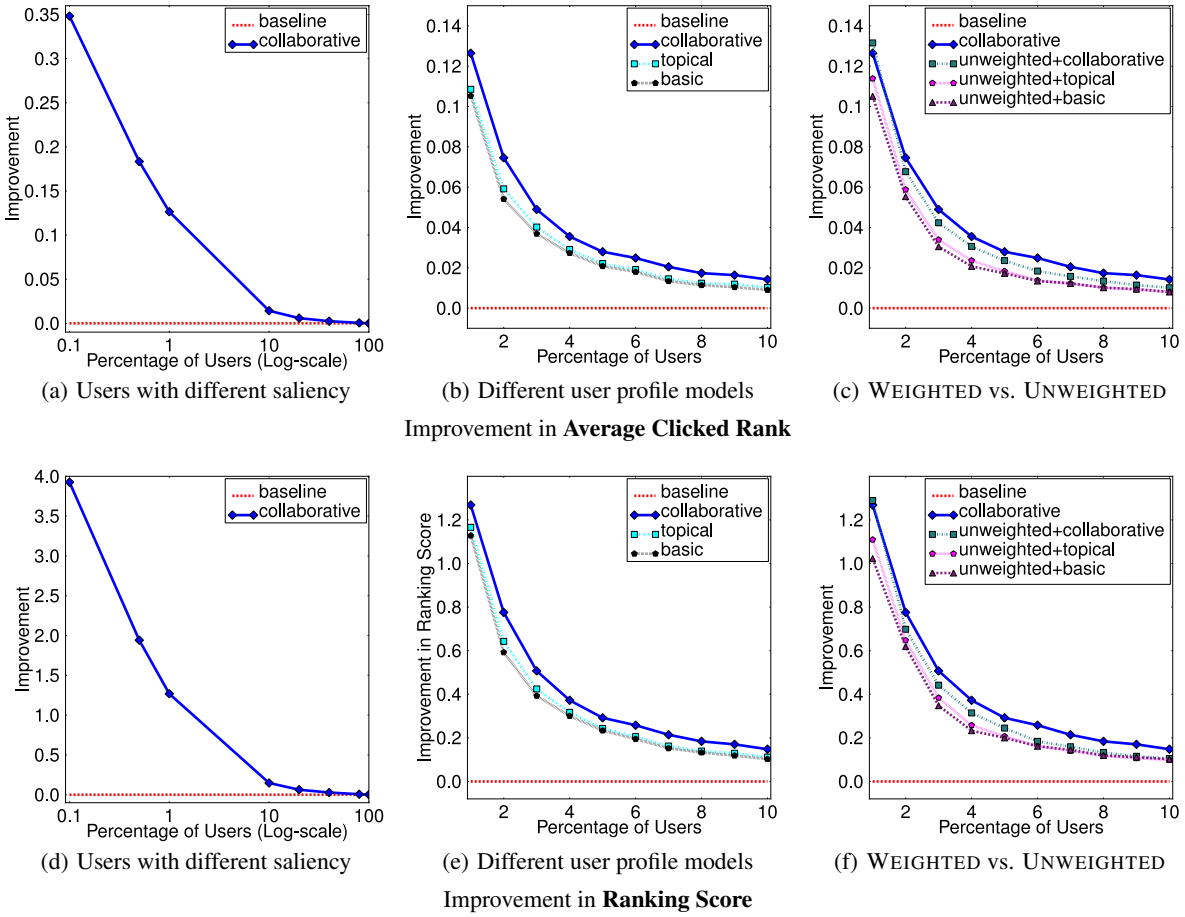
Among different user profile models, BASIC is more robust than COLLABORATIVE when we consider less salient users. If we fix the user profile model to be BASIC, and compare different pair-generation methods, CSA and LCAA beat the baseline more significantly than LCSA.

4.3.2 *Improvements for high-saliency users*

One reasonable setting is to enable comprehensibility-based personalization only for users with higher Q_u. The method (e.g., BASIC) that consistently beats the baseline for all users does not necessarily achieve the best performance for the high-saliency ones. Indeed, if we focus on the top 10% most salient users, WEIGHTED-LCAA +COLLABORATIVE achieves the biggest improvement over the baseline. We now examine this in more details.

Figure 3 summarizes the improvements in both average clicked rank and ranking score for a variety of settings. Figures 3(a) and 3(d) show the performance of WEIGHTED-LCAA +COLLABORATIVE. For the top 0.1% users, the improvement reaches 0.35 in average clicked rank, and almost 4.0 in ranking score. (Recall the performance of the strong baseline is around 3.6 in average clicked

[10]Note that a memory-based method will need to record all (user, query, clicked URL) pairs, and our method uses a much more compact representation. But an in-depth exploration of the performance tradeoff for this purpose is beyond the scope of this paper.



(a) Users with different saliency (b) Different user profile models (c) WEIGHTED vs. UNWEIGHTED

Improvement in **Average Clicked Rank**

(d) Users with different saliency (e) Different user profile models (f) WEIGHTED vs. UNWEIGHTED

Improvement in **Ranking Score**

Figure 3: Performance analysis focusing on high-Q_u users: comparing WEIGHTED-LCAA +COLLABORATIVE with its variations.

Table 3: Paired-t test for different methods against the baseline, computed for top 10%, 50%, and all users. We mark t-test results as follows: *($p < 0.05$), **($p < 0.01$), *($p < 0.001$)**

Method			10%	50%	100%
WEIGHTED	CSA	BASIC	***	***	***
		TOPICAL	***	***	***
		COLLABORATIVE	***	***	
	LCSA	BASIC	**	*	*
		TOPICAL	***	**	**
		COLLABORATIVE	***	***	
	LCAA	BASIC	***	***	***
		TOPICAL	***	***	***
		COLLABORATIVE	***	***	
UNWEIGHTED	CSA	BASIC	***	***	***
		TOPICAL	***		
		COLLABORATIVE	***		
	LCSA	BASIC	**		
		TOPICAL	**		
		COLLABORATIVE	***		
	LCAA	BASIC	***	**	**
		TOPICAL	***		
		COLLABORATIVE	***		

rank and about 74.4 in ranking score.) As expected and observed previously, performance degrades as we include less salient users.

We compared the improvement achieved by this configuration against other configurations using t-test. Keeping the same pair-generation technique (LCAA), it statistically significantly ($P < 0.05$) outperforms all other variations. However, its differences compared with WEIGHTED-CSA +COLLABORATIVE and WEIGHTED-LCSA +COLLABORATIVE are not statistically significant. At least for the top 10% users, different user profile models and the weighting scheme have more impact on the performance.

We now examine the effect of varying only the user profile model and only the weighting scheme over the best model, for the most salient 1% to 10% users.

Different user profile models

In contrast to our observations in Section 4.3.1 (where COLLABORATIVE was found to be less robust than BASIC when compared to the baseline for all the users), for the top 1%-10% users, COLLABORATIVE outperforms both BASIC and TOPICAL. Fixing the other parameters to WEIGHTED-LCAA, we plot the improvement achieved by using different user models (Figures 3(b) and 3(e)). While TOPICAL performs only slightly better than BASIC, COLLABORATIVE consistently outperforms both (statistically significant at $P < 0.05$ using t-test for top 10% users). This validates the effectiveness of modeling topic-specific comprehensibility preferences, as well as the necessity to employ collaborative filtering techniques to combat the sparseness in historical observations.

Different weighting schemes

Here we perform a direct comparison between WEIGHTED and UNWEIGHTED, and our findings reinforce the observations made in Section 4.3.1. As shown in Figures 3(c) and 3(f), our best configuration outperforms all other UNWEIGHTED variations using LCAA (statistically significant at $P < 0.05$ using the t-test for top 10% users). Interestingly, the difference between COLLABORATIVE and BASIC in the UNWEIGHTED setting is also less pronounced when more low-Q_u users are included (e.g., compare Figures 3(b) and 3(c) at the higher percentage of users).

4.4 Further analysis

In Section 4.3, we focused on performance break-down by user saliency. In this section, we examine the performance observed when using other ways to partition the data (measured at top 10% users using WEIGHTED-LCAA +COLLABORATIVE).

Improvement by topics The following table shows the improvements over the baseline in different topics.

Topic	Ranking Score	Average Rank
hobbies & interests	0.3550	0.0325
arts & entertainment	0.3517	0.0327
real_estate	0.2868	0.0302
society & culture	0.2094	0.0181
business & financial_services	0.1845	0.0195
technology & electronics	0.1519	0.0153
sports & recreation	0.1375	0.0155
employment & career	0.1038	0.0088
health & wellness	0.0778	0.0073
professional_services	0.0744	0.0059
education	0.0512	0.0065
retail	0.0426	0.0021
automotive	0.0362	0.0063
science & nature	0.0338	-0.0003
green	0.0336	-0.0000
home & garden	0.0258	0.0019
travel	-0.0521	-0.0008

In general, the topics with larger improments tend to be informational in nature (e.g., real estate, society & culture, business & financial services, and technology & electronics). On the other hand, for topics that are more transactional in nature (e.g., travel, home & garden), where comprehensibility may have less importance for users' information needs, we did not achieve similar improvements.

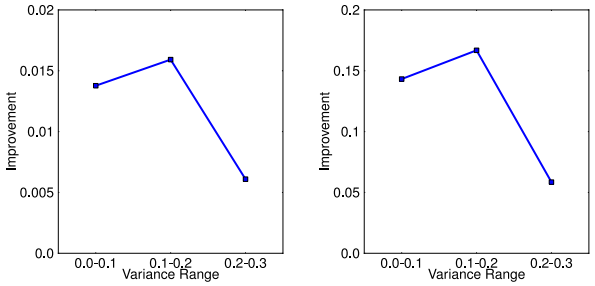
Per-query content variance For search results of a query, we can compute the variance in S_c. Figure 4 shows the improvement for different buckets of variance values. Most of the pages have a variance smaller than 0.3, and the plot focused on 3 buckets. The improvement increases as the variance increases from 0.1 to 0.2. This conforms to our intuition that comprehensibility-based personalization is more effective when the search results actually vary a lot in their comprehensibility. Interestingly, as the variance further increases to 0.3, the improvement decreases. This could happen because the higher variance in S_c can be partially attributed to topical differences, where the topical-relevance-based ranking needs to play a bigger role.

In our future work, we plan to incorporate both the query topic and the content variance in S_c to further improve our model.

5. EXPERIMENTS ON ANSWERS

We conducted a preliminary study on the Yahoo! Answers dataset. The goal here is to rank the answers posted to a question, based on the asker's personal comprehensibility preference.

Experimental setup To evaluate a given ranking, we simply measure the position of the best answer chosen by the asker in that ranking. Unlike the search data, there is no native ranking provided on Yahoo! Answers to use as a baseline or in rank combination.



(a) Average Click Rank (b) Ranking Score

Figure 4: Improvement vs. variance in S_c.

We experimented with the following systems:

- Random baseline: answers were ranked in random order. This baseline allows us to study whether comprehensibility preferences have any signal in predicting the best answer.
- Majority baseline: we take the preference of the majority of askers and does not allow personalization. On answers data, we observed $P_u > 0.5$ for an average user (i.e., a preference for harder content), thus we rank answers in decreasing S_c for all askers. This can be viewed as a proxy of a quality-based ranking since very low S_c could correlate with low-quality content in this domain.
- Our model: for answers data, we only have one pair-generation technique. We used the BASIC user profile that takes WEIGHTED preference pairs. As there is no native ranking to combine with, we simply rank the answers by increasing S_c if $P_u < 0.5$, and by decreasing S_c otherwise.

Results The table below summarizes the performance. We show the rank of the gold-standard best answer for different percentage of users (sorted in decreasing order by Q_u). That is, **lower** values indicate **better** performance. Shown in brackets are paired t-test results versus the random and the majority baseline, respectively: *($p < 0.05$), **($p < 0.01$), ***($p < 0.001$).

Fraction of users	Random	Majority	Our model
5%	3.375	2.947	2.895 (***, ***)
10%	3.596	3.096	3.079 (***, ***)
100%	4.525	4.093	4.149 (***,)

Both the majority baseline and our model beat the random baseline, and the improvements are highly significant. For users with lower saliency (i.e., 100%), the majority baseline, which may partially approximate answer quality, performs better. But for the top 5% and top 10% most salient users, our model is significantly better than the majority baseline. This indicates that modeling *personalized* comprehensibility preferences is helpful in identifying the best answers for different users.

6. DISCUSSION

6.1 User Preference in Search Data

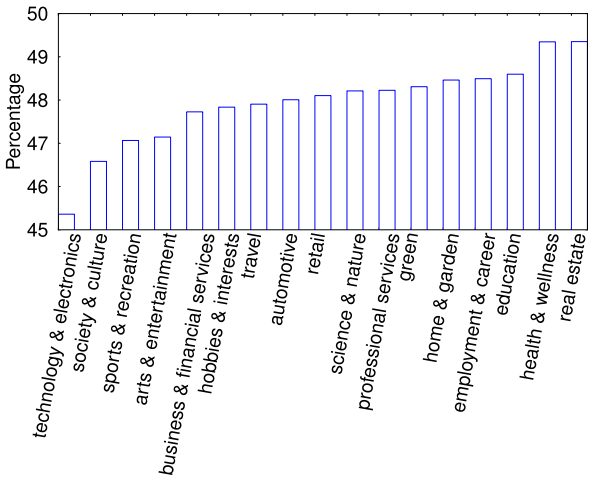
Across all users, P_u averages at 0.493 (using the BASIC method). [11] That is, among the search results returned to their queries, on average users have a slight preference towards easier content. We now

[11]Through out this study, we use LCAA unless specified otherwise.

address the following question: if we partition the data in different ways, will we consistently observe more users preferring easier content?

In what follows, we partition the users by their interests in different topics, as well as by gender and age. We rank the users by Q_u, and focus on the 60% most salient users (whose comprehensibility preferences are most pronounced). For each data partition, we report the percentage of users with $P_u > 0.5$ (denoted as R_h). The higher R_h is, the higher percentage of users prefer harder content. (Recall that the preference encoded in P_u is relative to the results returned for a given query, rather than an absolute preference for easy or hard content.)

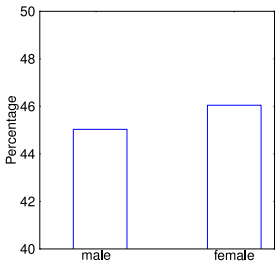
The figure below shows R_h for different topics. We observed a smaller percentage of users who prefer harder content in "technology & electronics", compared to topics like "health & wellness". We conjecture that for personally more important health-related content such as in "health & wellness", more people are willing to put in the effort to read more sophisticated content. (Alternatively, it could be due to search engine returning overly complicated results in "technology & electronics".)



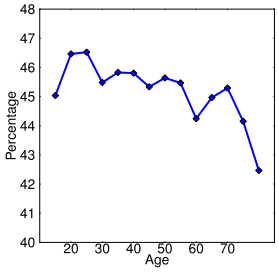
We also conducted a study on users whose demographical information is available. The figure below shows how R_h changes with gender and age. There is a slightly larger percentage of female users who prefer more sophisticated content (which could be partially correlated with different topical interests between the two genders). For different age groups (binned at 5 year increments), R_h peaks around the age of 25. In comparison, there is a smaller percentage of users preferring harder content among teenagers and seniors. In future work, such demographic information can be used to provide a better prior for users with insufficient data.

6.2 Discussion of design choices

In this work, we build user profiles using pairwise preferences (Section 2.2.1). In a preliminary study on search data, we experimented with an alternative approach, where we estimated the user's reading proficiency based on the average comprehensibility score of the (search result) pages she clicked on. We trained a variety of classifiers on a smaller-scale dataset, but did not observe any improvement over the baseline. We believe the model based on preference pairs is better suited for extracting comprehensibility-related signals from a search click log, where topical preferences and comprehensibility preferences may be conflated. Note that our framework allows us to build user profiles while accounting for



(a) Gender Study



(b) Age Study

position bias (Section 2.3). It also assigns different weights to preference pairs to focus more on pairs with a smaller difference in topical relevance. Perhaps more subtly, P_u captures user preferences relative to the search results for a given query, and does not compare comprehensibility scores of results from different topics, which may differ widely (Figure 1). Thus, if a user tends to formulate queries that trigger easier content than what he needs, he would tend to click on the harder results, which would lead to a higher P_u, even if the results clicked by the user do not necessarily receive a high S_c in terms of absolute value.

7. RELATED WORK

There are several bodies of prior research that are relevant to our study, namely, those focusing on text readability, personalized content selection, and collaborative filtering.

Text readability has been long studied in linguistics, and a number of (manually-tuned) readability indices have been proposed, such as the Gunning FOG Index [15], Flesch-Kincaid Index [23], Coleman-Liau Index [11], and others [21, 24]. Recent research employed statistical methods to quantify text readability. Collins-Thompson and Callan [13] predicted reading difficulty using statistical language modeling, which they showed to be superior to classical readability indices on a Web corpus. Schwarm and Ostendorf [29] used Support Vector Machines and statistical language models. Heilman et al. [16] combined lexical and grammatical features to quantify readability of first and second language texts. Pitler and Nenkova [26] employed readability measures to predict text quality. Bendersky et al. [3] formulated the notion of document quality, which captures text readability, layout, and ease of navigation, and showed it to be beneficial for document ranking. Several studies proposed ways to quantify answer quality in CQA forums [1, 32].

Personalized search was the focus of many studies that modeled users' interests in terms of topical relevance [34, 22, 9]. Dou et al. [14] studied different personalized search methods using click logs, and found the memory-based method to perform best. Teevan et al. [35] explored the utility of personalization for different types of queries. Recently, White et al. [37] argued that search experience can be improved by modeling user's domain expertise.

Arguably, the most related prior work is that by Collins-Thompson et al. [12], who studied personalization of Web search results by reading level. One key difference between our work and theirs is that they explicitly modeled 12 reading levels corresponding to the 12 US school grades. In contrast, we devised a supervised comprehensibility classifier based on the comparison between Simple English and regular English Wikipedia, as explained in Section 2.1. Our experiments also go beyond Web search, as we include results on Yahoo Answers as well. Finally, we employ

collaborative filtering techniques to alleviate data sparsity. In their follow-up work, Kim et al. [20] jointly model reading level and topic distribution.

Collaborative Filtering There is a vast body of work in collaborative filtering, and a comprehensive survey is not possible here. Most relevant to our work is the Maximum-Margin Matrix Factorization approach, which has been shown to be highly effective in recent work [28, 30, 2]. Cao et al. [8] formulated a collaborative ranking model for analyzing search click logs in a way that accounts for the position bias and alleviates data sparsity. Sun et al. [31] employed the (user, query, page) tensor built from click logs to improve personalized search.

8. CONCLUSION

We developed a unified framework for personalized content selection using text comprehensibility. We showed that modeling text comprehensibility can significantly improve content ranking, in both Web search and a CQA forum. We built a comprehensibility classifier to predict the sophistication level of a given text, and used it to construct topic-specific models of users' reading proficiency. Our approach is based on implicit user feedback, using search click logs or best answer tagging in CQA forums. We evaluated a number of strategies for extracting users' comprehensibility preferences from their historically-observed clicks, and found that manipulating pairwise preferences is far superior to merely averaging absolute comprehensibility scores across different topics. We also found that weighting preference pairs by position distance is beneficial, and also validated the effectiveness of collaborative filtering techniques in tackling data sparsity, which is inherent in modeling users' topic-specific preferences.

We believe that modeling text comprehensibility holds a lot of promise for personalizing content selection beyond topical relevance. Here we developed a text comprehensibility classifier by comparing pairs of Web pages. An immediate extension to this work would be to develop a comprehensibility-aware classifier of answers quality. In our future work, we plan to develop more sophisticated comprehensibility classifiers, as well as incorporate the difficulty of queries and questions into our framework.

Acknowledgment

We thank Oliver Chapelle, Fernando Diaz, and Lihong Li for discussions on evaluation metrics. We thank the anonymous reviewers for helpful suggestions.

9. REFERENCES

[1] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge sharing and Yahoo Answers: everyone knows something. In *WWW*, 2008.

[2] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *KDD*, pages 19–28, 2009.

[3] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of web documents. In *WSDM*, 2011.

[4] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, 1998.

[5] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[6] A. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. Zhang. Robust classification of rare queries using web knowledge. In *SIGIR*, 2007.

[7] A. Broder, E. Gabrilovich, V. Josifovski, G. Mavromatis, D. Metzler, and J. Wang. Exploiting site-level information to improve web search. In *CIKM*, 2010.

[8] B. Cao, D. Shen, K. Wang, and Q. Yang. Clickthrough log analysis by collaborative ranking. In *AAAI*, 2010.

[9] M. J. Carman, F. Crestani, M. Harvey, and M. Baillie. Towards query log based personalization using topic models. In *CIKM*, 2010.

[10] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW*, pages 1–10, 2009.

[11] M. Coleman and T. L. Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60:283–284, 1975.

[12] K. Collins-Thompson, P. N. Bennett, R. W. White, S. de la Chica, and D. Sontag. Personalizing web search results by reading level. In *CIKM*, pages 403–412, 2011.

[13] K. Collins-Thompson and J. P. Callan. A language modeling approach to predicting reading difficulty. In *HLT-NAACL*, pages 193–200, 2004.

[14] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW*, 2007.

[15] R. Gunning. *The Technique of Clear Writing*. McGraw-Hill, 1952.

[16] M. J. Heilman, K. Collins-Thompson, and J. Callan. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *HLT-NAACL*, 2007.

[17] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[18] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, pages 154–161, 2005.

[19] T. Kanungo and D. Orr. Predicting the readability of short web summaries. In *WSDM*, pages 202–211, 2009.

[20] J. Y. Kim, K. Collins-Thompson, P. Bennett, and S. Dumais. Characterizing web content, user interests, and search behavior by reading level and topic. In *WSDM*, 2012 (forthcoming).

[21] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom. Derivation of New Readability Formulas for Navy Enlisted Personnel. Technical report, NTIS, Feb. 1975.

[22] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *WSDM*, pages 25–34, 2011.

[23] G. McClure. Readability formulas: Useful or useless. *IEEE Transactions on Professional Communications*, 30:12–15, 1987.

[24] G. H. McLaughlin. SMOG grading—a new readability formula. *Journal of Reading*, 12(8):639–646, 1969.

[25] D. Metzler, J. Novak, H. Cui, and S. Reddy. Building enriched document representations using aggregated anchor text. In *SIGIR*, 2009.

[26] E. Pitler and A. Nenkova. Revisiting readability: a unified framework for predicting text quality. In *EMNLP*, 2008.

[27] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *KDD*, 2005.

[28] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

[29] S. E. Schwarm and M. Ostendorf. Reading level assessment using support vector machines and statistical language models. In *ACL*, pages 523–530, 2005.

[30] N. Srebro, J. D. M. Rennie, and T. S. Jaakola. Maximum-margin matrix factorization. In *NIPS*, 2005.

[31] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen. CubeSVD: a novel approach to personalized web search. In *WWW*, 2005.

[32] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online QA collections. In *ACL*, 2008.

[33] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts. Information re-retrieval: repeat queries in yahoo's logs. In *SIGIR*, 2007.

[34] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR*, 2005.

[35] J. Teevan, S. T. Dumais, and D. J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *SIGIR*, pages 163–170, 2008.

[36] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola. CoFiRank, Maximum Margin Matrix Factorization for Collaborative Ranking. In *NIPS*, 2007.

[37] R. W. White, S. T. Dumais, and J. Teevan. Characterizing the influence of domain expertise on web search behavior. In *WSDM*, 2009.