Language Intent Models for Inferring User Browsing Behavior

Manos Tsagkias*
ISLA, University of Amsterdam
Amsterdam, The Netherlands
e.tsagkias@uva.nl

Roi Blanco
Yahoo! Research
Barcelona, Spain
roi@yahoo-inc.com

ABSTRACT

Modeling user browsing behavior is an active research area with tangible real-world applications, e.g., organizations can adapt their online presence to their visitors browsing behavior with positive effects in user engagement, and revenue. We concentrate on online news agents, and present a semi-supervised method for predicting news articles that a user will visit after reading an initial article. Our method tackles the problem using language intent models trained on historical data which can cope with unseen articles. We evaluate our method on a large set of articles and in several experimental settings. Our results demonstrate the utility of language intent models for predicting user browsing behavior within online news sites.

Categories and Subject Descriptors

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

General Terms

Algorithm, Experiment

Keywords

Online news, article intent models, user, browsing, behavior

1. INTRODUCTION

Social media has changed the way news are produced and consumed, with well established news publishers having lost large share of their readers. The continuous decline in readership is also reflected in revenue, urging news publishers to seek new ways for monetizing their news articles. One of the many ways to do so is to increase the amount of time users spend on a news site. Central in achieving an increased user dwelling time within a site's property is the concept of *user engagement* [3], or quality of the user experience with an emphasis on the positive aspects of the interaction. Research on this area suggests that enhancing a web page with contextual information has positive impact on user engagement, and it has led to the development of a range of systems that address this phenomenon [9, 19]. A key ingredient here is to discover the right context for a web page, especially in a setting where user goals might change after the user visits the web page and new content is added continuously.

Context discovery can be cast as a task in the realm of semantics, personalization, or collaborative filtering, with the first being the most typical interpretation. Contextual information is drawn from a knowledge base built on a single source, e.g., news, Wikipedia or the web page's source domain, and typically remains relatively static over time. In the news domain, there is need for systems able to adapt, and discover the right sources of context in a dynamic fashion. As a working example, think of one news article announcing a forthcoming football game, and one reporting on the results of the game. In the first article a user might be interested in seeing information about the teams' setup, whereas in the second in the game highlights. Other examples are news articles reporting great natural disasters, e.g., Haiti's earthquake, or the tsunami in Japan, where users might want to see information about emergency services, the Red Cross, or weather reports.

Browsing behavior far outweighs direct search engine interaction as an information-gathering activity [26]. Given so, our focus is to recommend websites as opposed to search results. In contrast with previous approaches [26] which employ contextual sources for modeling a particular user's interests, we focus only on textual features extracted from the text of the articles browsed by the user and queries issued within a query session.

The main focus on in this paper is the *temporal context discovery* task: *For a given news article, and optionally a user, the system needs to discover webpages that the user is likely to visit after reading the article.* The task is challenging due to data sparsity issues that arise from the inherent volatility of the news domain, and the broad range of user intents, which lead to a heavy tailed distribution of user destinations after they visit a news article. To quantify this claim, we conducted an exploratory experiment. From a set of logical sessions extracted from query logs we identified web pages that are news articles, and classified them into categories, e.g., Science, Business, Entertainment. For each article we recorded the internet domains of the web pages that users visited after reading a news article, and assigned

[*] This work was conducted during a three–month internship at Yahoo! Research Barcelona.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
Copyright 2012 ACM 978-1-4503-1472-5/12/08... $15.00.

these domains to the article's news category. We also record the popularity of a domain per category by counting the number of visits to the domain from articles in that category. Fig. 1 illustrates our findings on users navigational patterns after reading web pages in Yahoo! News. Red circles represent news categories, and white to blue shaded circles represent domains (white denotes least popular, and dark blue highly popular). News categories are laid out in space based on how they are connected to the domains. This results in an inner circle of news categories which forms a perimeter within which lie domains shared by these news categories, and outside of it are domains mostly unique to each category. The outer red dot perimeter includes categories that don't share common domains, i.e., Wikipedia, Yahoo! news, search engines, but share one or two rather "unique" domains attached to another category. Our findings suggest that there is a distinct set of domains where people navigate to depending on the category of the news article they read, forming a heavy tailed distribution of user navigational patterns. These challenges, i.e., data sparsity, and the cold start problem, restrain us from training robust recommendation models for articles that appear in user sessions, and make online recommendation virtually impossible for articles with no record in user sessions (we have to wait for at least one user trace).

Our approach to overcome these limitations is to cast the temporal context discovery task as an information retrieval problem, and develop methods for modeling user browsing intent in a query. These methods infer the user navigation patterns, by mapping the current article a users reads to a query into *intent space* which represents the content of articles likely to be clicked after the current one. This way, we aim to answer both challenges raised above; First, modeling article intent as proxy for user browsing intent helps to smooth out data sparsity issues. Second, modeling article intent allows for making predictions for unseen articles via the article intent space. Our experiments show that using text-based features and query session trails are able to achieve good performance for the task of temporal context discovery.

We envisage our methods to have concrete applications in enhancing user experience and engagement via dynamic link suggestion, result recommendation, and personalized web content optimization. Such applications can prove valuable to news agents, publishers, and consumers. News providers can leverage this information to generate focused recommendations via links to their consumers. In turn, consumers can save time completing their goal as relevant hyperlinks, or snippets from likely to visit web pages, and ads can be displayed on the same web page as the article.

The main contribution of this work is a model which uses historical data for making real-time predictions about the browsing patterns of users, for which little information is needed to be available at prediction time.

The rest of the paper is organized as follows. In the following section, Section 2, we present related work. In Section 3 we describe the problem definition, in Section 4 we outline our approach to the problem, in Section 5 and 6 we present our modeling and retrieval approaches, in Section 7 we describe our experimental setup, in Section 8 we report on results, in Section 9 we further discuss our findings, and in Section 10 we conclude.

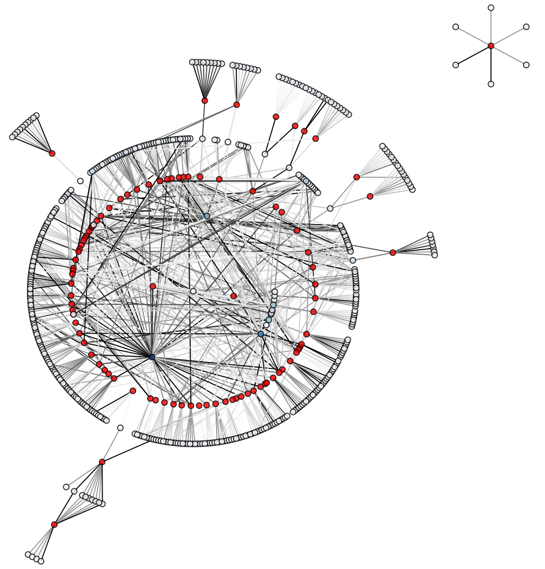

Figure 1: News categories show distinct patterns of where users navigate next after reading a news article. Red circles denote news categories, blue shaded circles denote internet domains; darker shades represent targets from many news categories. The two dark blue domains in the middle correspond to Wikipedia, and Yahoo! News.

2. RELATED WORK

The increased availability of query sessions coming from the logs of search engines has grown a research area that deals with studying, mining and making use of trails and user actions to improve search [12]. Search trails are sequences starting on a query and ending on a destination page, with a series of intermediate pages browsed by the user. For instance, these trails are a useful signal in order to learn a ranking function [1, 4] or to display the trails directly to the user [23] to help in the information seeking process. These approaches try to employ the query session information as implicit feedback in order to incorporate it into the ranking process. In contrast, our approach is targeted towards news article recommendation, using a mixture of the trail pages as a query.

There is an increased attention on techniques to identify the intent behind the query; this is one main challenge for modern search engines [7]. The first query classification was presented by Broder [6], in which queries were labeled as transactional, navigational or informational. However there are more sophisticated approaches, which take into account the multidimensionality of queries [10]. Approaches to classify web queries are mostly based on click-through data [15]. These methods are a key factor into identifying the real user's goals, and their applicability ranges from personalizing search results to predicting ad click-through [2], or search result diversification [8, 21].

Guo et al. [11] look at intent-aware query similarity for query recommendation. Intent is identified in search result snippets, and click-through data, over a number of latent topic models. Our approach differs in that the intent is modeled to capture the characteristics of the news domain and we do not recommend queries, but rather news articles. We also do not attempt to classify queries into a predefined set of categories, but rather we use the content of the clicked articles as an extended representation of the user intent.

Finally, there exists other possibilities for article recommendation, for instance those based on the explore-exploit framework like the one of Li et al. [16]. Those approaches require a significant amount of click-through traffic and in general are content-agnostic, using as similarity features clicks shared between users.

3. PROBLEM DEFINITION

We cast the problem of *temporal context discovery* as follows. Given a document $\alpha \in \mathcal{A}$, and a set of user sessions $\mathcal{T}$, find a ranked list of documents $\{\alpha\}_i \subseteq \mathcal{A}$ that a user is likely to read after reading α. The session set is defined as

$$\mathcal{T} := \{(q, \alpha, \ldots, o_j)\},$$

where $q \in Q$ represents a query, $\alpha \in \mathcal{A}$ is a document, and o_j is either a document $\alpha_j \in \mathcal{A}$ or another query $q_j \in Q$.

In this work, we reduce the complexity of the problem in two ways. First, we only focus on the news domain. In this respect, documents in $\mathcal{A}$ are news articles, and the majority of user queries we deal with is of informational type [6]. Second, we focus on methods that use only textual information derived from the articles' content. We do not use additional information from query logs or the web graph as signals for ranking, e.g., time spent on each document, hyperlinks between the documents. In particular, we omit the use of hyperlinks because they are not always present in news articles (see Section 1), and can potentially bias the evaluation as they reduce the recommendation space to the articles linked by the current one.

The problem at hand is similar to that of recommending similar articles to the article currently being read. However, in the current setting, the system has to probe user intent and recommend articles not only based on the user's cognitive model at the query issue time, but also on the changes that occur in their cognitive model after reading a news article. This requirement asks for an approach beyond recommending articles which are semantically or topically similar to α. In this setting, we face a challenging task as the recommendations have to adapt quickly and incrementally to reflect the ongoing process in the user's cognitive model.

To make things more tangible, consider a search session from a user υ that consists only of queries, and news articles. These query sessions [5] are records of the queries and the actions of the users of search engines, and they contain latent information about their interests, preferences, and behaviors. Let two users υ_1, and υ_2 issue the same informational query q to a search engine, and then click on a retrieved news article, possibly read it, then return to the search results, and click on another article. In the process, they may choose to refine their query with the current state of their cognitive model which has now possibly changed after visiting the retrieved news articles. This iterative process will generate the following traces:

$$\begin{aligned} \upsilon_1 &:= q_1 \rightarrow \alpha_1 \rightarrow \alpha_2 \rightarrow q_2 \rightarrow \alpha_3 \rightarrow \cdots \rightarrow \alpha_{\upsilon_1} \\ \upsilon_2 &:= q_1 \rightarrow \alpha_3 \rightarrow q_2 \rightarrow \alpha_1 \rightarrow \alpha_2 \rightarrow \cdots \rightarrow \alpha_{\upsilon_2} \end{aligned}$$

In these traces we see user υ_1 issuing a query, then visiting article α_1, then going back to the results page and selecting another article α_2. After visiting α_2, υ_1 decided to refine their query and issued q_2, and continued visiting other articles. A similar process occurs for user υ_2, however, the order υ_2 visits the article is different, and also, the position within the trace of the second query is different. The temporal context discovery task is defined as: predict $\alpha_2, \ldots, \alpha_\upsilon$ given q and α_1 for a user υ.

4. APPROACH

Our approach is to estimate a query model $\hat{q}$ that reflects user's browsing intent, namely, what article the user is likely to read after clicking α_1 and given their query trail τ_k. The rationale is that when $\hat{q}$ is submitted to an index of articles, a retrieval system will return documents that are likely to be read from user υ_k. To this end, our efforts are concentrated on modeling the query $\hat{q}$.

A key aspect here is to derive a robust method for modeling the user's browsing intent. We build on the intuition that user intent on a news article α can be captured by training models on the content of the articles that follow α. The rationale is that these models should capture the relation between content and patterns of user behavior using the query sessions. The query sessions define links between each article in the intent space. We call these models *article intent models* (AIMs). Fig. 2 illustrates this idea, and the steps we take afterwards.

Articles for which the system will make predictions do not necessarily exist in the news article pool, or have been recorded in user sessions which leaves them without intent models. We account for this issue by building on the assumption that similar articles lead to similar user traces. This way, articles that do not occur in user sessions are assigned the intent model of the most similar article that has one. This idea also helps assigning intent models to previously unseen articles, and allows coping with an expanding pool of articles.

With the article intent models in place, we estimate the query $\hat{q}$ using information from either the content of the article, its corresponding intent model, and a mixture of both. For the latter case, we derive several weighting methods which are explained in Section 5.4.

A retrieval system based on a widely used, state-of-the-art information retrieval method receives the query $\hat{q}$ and returns a ranked list of news articles. We consider two options for this. The first option submits the query to an index of articles, while the second option issues the query to an index of intent models, the ranked list of which is mapped to news articles.

Relevant articles are deemed those that follow α in user sessions. In order to ensure that the user has read the article in question, we discard articles in which users spent less than 30 seconds [14]. The system is evaluated on whether it manages to retrieve these relevant articles in early positions.

5. MODELING

We present the methods for addressing the steps involved in our approach: (a) Model the news article, (b) model article intent, and (c) model queries, which we consider as a mixture model of the first two steps.

We start with a pool of news articles $\mathcal{A} := \{\alpha_1, \ldots, \alpha_N\}$, where N is the size of the pool, and a list $\mathcal{T} := \{\tau_1, \ldots, \tau_K\}$

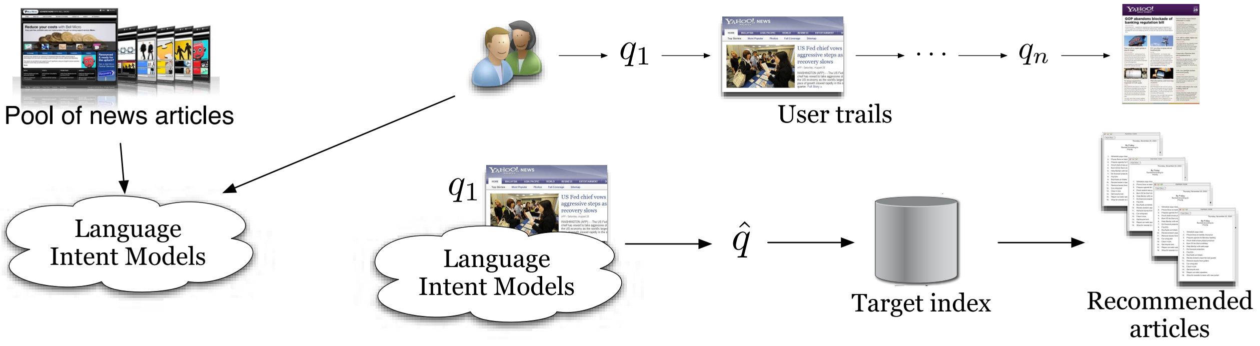

Figure 2: Our approach for recommending articles based on user intent.

Table 1: Description of the main symbols we use.

Symbol	Gloss
$\mathcal{A}$	Pool of news articles in the database
$\mathcal{T}$	Pool of user traces
τ_k	k trace in $\mathcal{T}$
υ_k	User identifier of trace k
q	Query
α_n	Article n in $\mathcal{A}$
w	token in a language model
${}_c\theta_n$	n-th article LM trained on content
${}_p\theta_n$	n-th article LM trained on persons
${}_o\theta_n$	n-th article LM trained on organizations
${}_l\theta_n$	n-th article LM trained on locations
${}_t\theta_n$	n-th article LM trained on time expressions
$\boldsymbol{\theta}_n$	n-th article language model vector
$\boldsymbol{\theta}_n^I$	n-th article intent model vector (AIM)
$P(w\|\theta_n)$	Probability of sampling w from n-th article LM
$P(w)$	A priori probability of sampling w
$n(w,\alpha_n)$	Frequency of w in n-th article
$sKL(\cdot)$	Symmetric KL divergence between α_1 and α_2
ξ	Weight parameter for LM linear combination

of user traces $\tau_k := (\upsilon_k, q, \alpha_1, \ldots, \alpha_{l_k})$, similar to the ones described in Section 3. K denotes the total number of user traces in our database, $k := \{1, \ldots, K\}$, υ_k is an anonymized user identifier, and l_k is the length of user trace k in clicks. Table 1 contains a list of symbols used throughout the paper.

5.1 Article models

Articles α_n are represented as language models drawn from ϕ different distributions, each one defined over its own event space. To reduce clutter, we refer to all ϕ language models as $\boldsymbol{\theta}_n$ to denote a vector of language models:

$$\boldsymbol{\theta}_n := \langle {}_1\theta_n, \ldots, {}_\phi\,\theta_n \rangle \,.$$

For achieving a rich representation of the textual content of the article, we focus on three families of sources for learning language models: (i) the unigrams of the news article, (ii) the named entities therein, and (iii) the time expressions mentioned in the article. We motivate the use of each source in turn.

Article content. The body of the news article itself is an important source of information for training language models that represent it [20, 24, 27], as witnessed from the successful previous work in probabilistic modeling for retrieval. We follow [18, 25] and use entire contents of article body, and title for training a unigram language model.

Named entities. A great majority of news refer to and discuss people, organizations, and locations. To this extent, we extract named entities, and train a language model per named entity type, i.e., persons, organizations, and locations. The rationale behind is that if an article focuses on a particular named entity, the named entity will occur many times in the article, resulting in a language model that emits this named entity with high probability.

Temporal expressions. Real world events are central to news reporting, and news articles connect the development of events through time expressions, e.g., “last week”, “in one month”. Using time expressions can help identify articles that discuss the same time period of an event [13]. We group time expressions into three classes, i.e., *past*, *present*, and *future* relative to the article’s publication date. A language model trained on time expressions consists of these three classes, and the probability of emitting a class is proportional to the number of time expressions that belong to this class.

For every article α_n in $\mathcal{A}$ we train language models from the three different sources we described above: the article content, the named entities, and the temporal expressions. We assume that each article is drawn from three multinomial distributions, each one defined over its own event space $\mathcal{E}$ of token occurrences w. We smooth the number of times a token w is present in the article using Dirichlet priors, thus defining the probability of a token w to be generated from the language model n as:

$$P(w|\theta_n) = \frac{n(w,\alpha_n) + \mu P(w)}{|\alpha_n| + \mu},$$

where $n(w,\alpha_n)$ is the frequency of token w in α_n, $|\alpha_n|$ is the article length in tokens, $P(w)$ is the a priori probability of w, and μ the Dirichlet smoothing hyper-parameter [27].

5.2 Article intent models

Now, we shift from the content space to the intent space through the user traces in $\mathcal{T}$. An *article intent model* (AIM) $\boldsymbol{\theta}_n^I$ intends to capture the original user intent by using the articles that the user reads after α_n as proxy. More formally, $\boldsymbol{\theta}_n^I$ is defined as the combination of the language models of the articles users browsed afterwards,

$$\boldsymbol{\theta}_n^I = \sum_{k=1}^{K} \sum_{i=j}^{l_k} \lambda(i)\boldsymbol{\theta}_i,$$

where j is the position of α_n in trace τ_k, and $\lambda(i)$ a weighting function dependent on the position of an article within τ_k. $\lambda(i)$ is likely to be a exponential decay function, however, due to the sparseness of the dataset we set it to be uniform over all article models.

Noise in query logs, along with data sparsity, i.e., the small number of articles users visit after reading a news article (see Section 7 for a description of our dataset) can lead to poor article intent models. To account for this effect, we describe a method for assigning more than one AIM to an article. We work as follows. First, we compute the pairwise similarity of all articles in the pool that have associated article intent models. Then, we assign each article α_n a vector V_n of tuples that consist of an article intent model along with the similarity scores of the article intent model's associated article:

$$V_n := \left\langle (1, \boldsymbol{\theta}^I_n), \ldots, (s_\nu, \boldsymbol{\theta}^I_\nu) \right\rangle,$$

where $(1, \boldsymbol{\theta}^I_n)$ denotes the article intent model associated with α_n, and $(s_\nu, \boldsymbol{\theta}^I_\nu)$ is the article intent model for α_ν which has s_ν similarity with α_n.

Intent models for unseen articles. In many situations, the system will need to map articles that do not exist either in $\mathcal{A}$ (e.g., new article is added) or in $\mathcal{T}$ (e.g., the article has no logged visits yet) to the intent space. Given that we define all the models over the same event distributions, the method builds on the hypothesis that users reading similar articles are likely to have similar intent, and therefore produce a similar user trace.

Let α_n be an article with no AIM associated with it, we want to find similar articles to α_n for which there exist AIMs. If the intent models are generated from an unknown data distribution $P(\boldsymbol{\theta}^I)$, our goal is to find a model $\boldsymbol{\theta}^I_n$ such that the marginal probability computed over the whole intent model space is maximized:

$$P(\boldsymbol{\theta}^I_n) = \int P(\boldsymbol{\theta}^I_n|\boldsymbol{\theta}^I)P(\boldsymbol{\theta}^I)d\boldsymbol{\theta}^I.$$

We approximate the integral using the finite set of intent models generated from the articles in $\mathcal{A}$:

$$\sum_{j=1}^{|\mathcal{A}|} P(\boldsymbol{\theta}^I_j)P(\boldsymbol{\theta}^I_n|\boldsymbol{\theta}^I_j).$$

There are several possibilities for selecting $\boldsymbol{\theta}^I_j$; we make the assumption that documents with similar language models have similar intent models, and therefore $P(\boldsymbol{\theta}^I_n|\boldsymbol{\theta}^I_j) \propto sim(\boldsymbol{\theta}_n|\boldsymbol{\theta}_j)$. The article index selected is the one that maximizes

$$j = \operatorname*{argmax}_{j\in\{0,\ldots,|\mathcal{A}|\}} (sim(\boldsymbol{\theta}_n|\boldsymbol{\theta}_j)), \quad (1)$$

and $\hat{\boldsymbol{\theta}}^I_n = \boldsymbol{\theta}^I_j$. The final intent model is interpolated with the original model as:

$$\boldsymbol{\theta}^I_n = \xi\hat{\boldsymbol{\theta}}^I_n + (1-\xi)\boldsymbol{\theta}_n,$$

where ξ is a parameter defining the weight of each language model.

In order to select the most similar intent model in Eq. 1 we create an index of all the language models generated from $\mathcal{A}$, and rank them using $\boldsymbol{\theta}_n$ as a query, with the standard symmetric KL-divergence as a ranking function (defined in Section 6).

5.3 Query models

The previous sections have presented our approach to modeling news articles, and article intent models. We move on how to use them for estimating a query $\hat{q}$ for a user υ_k. A straightforward way is to make the simplifying assumption that both the query issued by the user, and the article α_n they first read are representative for the user's intent. Formally, the estimated query can be written as:

$$\hat{q}^{ART} := \rho_q\theta_q + (1-\rho_q) \sum_{i\in\{c,p,o,l,t\}} \kappa_i \cdot {}_i\theta_n, \quad (2)$$

where ^{ART} denotes the estimation of the query using article models, and the original query model, ρ_q stands for the weight assigned to the query language model θ_q, κ_i denotes the weights for the different article language models ${}_i\theta_n$, i.e., content, named entities, temporal expressions. This user model operates in the vertical dimension in the sense that assumes that people are interested in reading similar articles to the one they first read, because, for example, they want to find more information about the topic of their interest. Although this assumption may stand true for certain users, it ignores the horizontal dimension, namely, users that want to read different aspects of the article they first read, or find information related to it but not directly.

We make the hypothesis that article intent models can serve this purpose, namely, model the user intent, for finding articles that users are likely to read afterwards. In this respect, we estimate the query $\hat{q}^{AIM}$ from article intent models associated with the article that the user visited first:

$$\hat{q}^{AIM} := \sum_{\nu}^{|V_n|} \beta_\nu \cdot \sum_{i\in\{c,p,o,l,t\}} \kappa_i \cdot {}_i\theta^I_n, \quad (3)$$

where ^{AIM} denotes the estimation of the query on article intent models, V_n is a vector with article intent models for α_n, and β_ν is a weight for each article intent model in V_n. The building block of these models depends on user trails extracted from query logs. Query logs are known to contain noise [22], and therefore using article intent models instead of article models may introduce topical shift, possibly degrading the quality of the list of suggested articles.

A natural way to tackle this problem is to model user intent as a mixture model of the user's query, the first article they read, and article intent models. We define a mixture model for modeling $\hat{q}$ as $\hat{q}^{ART+AIM}$:

$$\begin{aligned}\hat{q}^{ART+AIM} &:= \beta_{inc}\, \hat{q}^{ART} + \hat{q}^{AIM} \\ &= \beta_{inc}\, \hat{q}^{ART} + \sum_{\nu=1}^{|V_n|} \beta_\nu \cdot \sum_{i\in\{c,p,o,l,t\}} \kappa_i \cdot {}_i\theta^I_n, \quad (4)\end{aligned}$$

where β_{inc} is the weight regulating the importance of the user's query, and the first read article.

5.4 Weighting schemes

A straightforward procedure for estimating the weights in Eqs. (2)–(4) is to use supervised learning. In particular, without imposing any restriction in the parameters, the model could assign one weight per article in every trace, which asks for significant amounts of training data. This requirement renders supervised learning unsuitable for an online setting, where the pool of articles expands continuously, or where training data is scarce. The approaches we describe next aim at overcoming this problem by producing the query mixture model $\hat{q}^{ART+AIM}$ without the need for training. They build on knowledge derived from the distribution of similarities between the $\boldsymbol{\theta}_n^I$ vectors. The hypothesis is that the semantic similarity between an incoming article and its intent models is a good surrogate for regulating weights in query mixture models. To this end, we describe several strategies that create query mixture models for an article model, using one or several article intent models. We separate the two cases of *one* and *many* article intent models because of the implications in weighting.

Merge. We begin with a simple approach, namely, assign no weights to the article or to its article intent models, but merge their contents before training language models for content, named entities, and temporal expression. This can be generalized for any number of article intent models we want to consider.

Pairwise. This technique considers mixture models for an article and its most similar article intent model. We assign the incoming article intent model weight $1 - s_\nu$, and the intent model the weight s_ν, where $0 \leq s_\nu \leq 1$ is the semantic similarity between them. We also try the reverse, namely, the incoming article weights $1 - s_\nu$, and the intent model s_ν. We refer to this second approach as *Pairwise-R.*

Average. When it comes to add more than one AIM to the mixture model, we face the problem what weight to assign to the incoming article. One way is to assume that the incoming article language model is an article intent model trained only on itself and therefore its weight is set to 1. Then, we enforce the constraint on weights to sum up to 1:

$$\beta_{inc} + \beta_1 + \cdots + \beta_\nu = 1,$$

where $\beta_{inc} = 1$, which transforms the weights to:

$$\beta'_x = \frac{\beta_x}{\beta_{inc} + \beta_1 + \cdots + \beta_\nu}.$$

Median. The previous approach makes the assumption that an article model is semantically identical to an article intent model. We try to smooth this assumption by weighting the incoming article proportionally to the set of its article intent models. The weights of similar article intent models show a broad range of values, therefore their median value can be a good indicator for weighting the incoming article. In other words, if the median value is high, then we give preference to the article intent models as they are likely to bear more information, and vice-versa, if the median value is low, then we give preference to the incoming article as it is likely to be more informative for retrieval. Formally:

$$\beta_{inc} = 1 - m(\{\beta_1 + \cdots + \beta_\nu\}),$$

where $m()$ is the median value.

6. RETRIEVAL

All the formulation presented insofar builds comparable model representations. In order to retrieve articles, represented by either their language or intent models, as a response to a query $\hat{q}$ we use the symmetric Kullback-Leibler divergence. This is, given two vectors of models $\boldsymbol{\theta_t}$ and $\boldsymbol{\theta_n}$ we compute a score as

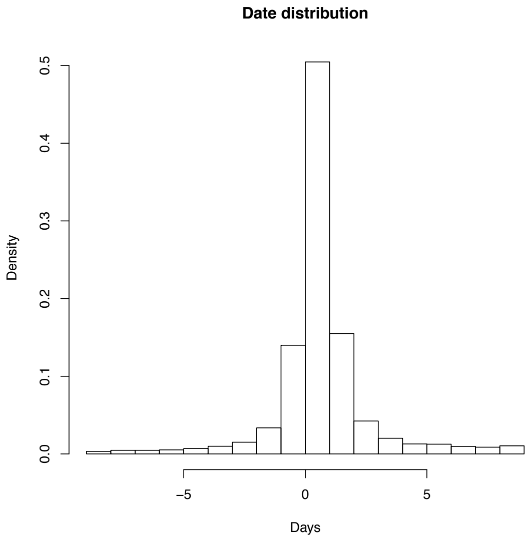

Figure 3: Distribution of the date difference in days between the articles users have clicked in a session, aggregated over all sessions. Positive difference indicates articles published prior to the input article. Showing differences less than 10 days for clarity.

$$\begin{aligned} score(\boldsymbol{\theta_t}||\boldsymbol{\theta_n}) &:= sKL(\boldsymbol{\theta_t}|\boldsymbol{\theta_n}) \quad (5)\\ &= \sum_{vc\in\{c,p,o,l,t\}} \left[\sum_w P(w|_{vc}\theta_t)\log\frac{P(w|_{vc}\theta_t)}{P(w|_{vc}\theta_n)}\right.\\ &\quad \left.+ \sum_w P(w|_{vc}\theta_n)\log\frac{P(w|_{vc}\theta_n)}{P(w|_{vc}\theta_t)}\right], \end{aligned}$$

where w is a token from the union of tokens in the respective language models.

In order to recommend articles, we need to rank $\hat{q}$ with respect to α_n. In this case $\hat{q}$ plays the role of $\boldsymbol{\theta_t}$ in Eq. 5 and $\boldsymbol{\theta_n}$ or $\boldsymbol{\theta_n^I}$ play the role of $\boldsymbol{\theta_n}$, when we consider the model of the article or its AIM respectively.

Temporal bias. Our ranking model assumes a uniform distribution over the likelihood of user preference on ranked documents. We examine whether this assumption holds by plotting the time difference of publication of articles that users visited after reading an initial article. Fig. 3 shows the user preference is biased towards articles published close to the first article they read. It has a strong peak at 0 days, rapidly decreasing in both sides, possibly due to the presentation bias in the search results. We model this phenomenon with the standard Cauchy distribution,[1] which introduces a bias towards articles visited shortly after the article at hand:

$$\lambda(i) = \frac{1}{\pi}\left[\frac{1}{(i-j)^2+1}\right].$$

[1] We experimented with modeling this distribution with a block exponential function, double exponential (Laplace distribution) and we found that among them using the Cauchy distribution gives the best retrieval effectiveness; see §9.

7. EXPERIMENTAL SETUP

In this section we describe our research questions, experiments, dataset and evaluation methodology. Our main research question we aim to answer is whether our query models can help in the task of temporal context discovery. We study this question in the following three dimensions:

Query modeling What is the effect in retrieval effectiveness when using our three query modeling approaches? What is the effect of temporal bias in the retrieval model?

Weighting What is the effect in performance of our weighting schemes?

Retrieval in intent space Can effectiveness be improved if we perform retrieval in the intent space instead of the article space?

To answer these research questions, we proceed as follows. First, we compare the three query models we presented in Section 5.3 which use either the query and the incoming article, or the article's intent model, or their combination. We study the effect of time in retrieval performance using retrieval models with and without temporal bias. Next, we focus on the weighting scheme for generating the query mixture models, and compare each of them. Finally, we change our index from articles to article intent models, and use our query models to retrieve article intent models which are then mapped to articles in a post-retrieval stage.

In Table 2 we list the alternatives we consider, along with their corresponding features. Runs that use only the article for modeling the query are denoted with ART, those using only article intent models AIM, and their combination $\mathrm{ART} + \mathrm{AIM}$. Query models on temporally biased retrieval modes have a superscript T, and different weighting methods are denoted in the subscript. For example, $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$ is model that uses both the incoming article and the article intent models on a temporally biased retrieval model, using the Median weighting method for generating the query mixture model.

7.1 Dataset

Our dataset consists of 14,180 Yahoo! News items published in February 2011, and a parallel corpus of query logs from Yahoo! Search. We apply several preprocessing steps. We extract named entities using the SuperSense tagger,[2] and time expressions using the TARSQI Toolkit.[3] The query stream is segmented into several sets of related information seeking queries, i.e., logical sessions using the technique in [5]. The logical sessions are pruned to contain only queries and articles that exist in our article dataset.

Our experiments include a training, and a testing phase. We use 75% of the logical sessions for training article intent models for 3,060 articles, and the remaining 25% as ground truth for 434 query test articles.

7.2 Evaluation

We assemble our ground truth as follows. From the logical sessions in the test set we consider the first user query and article in the session as our input, and consider every following article as relevant to this input. This process results in a ground truth of 511 relevant documents (max/min/avg: 3/1/1.18 documents).

In our experiments we work as follows. Given the user query and the article, we generate a query $\hat{q}$ with our methods which we then use to retrieve articles from either an index of articles, or article intent models. We treat the user query and the input article equally as in Eq. (2). For query mixture models, we consider one article intent model, the most similar to the input article.

For our experiments we use the Indri framework [17]. We set the weights in an independent held-out data-set as follows: for named entities to 0.1, for temporal expressions to 0.1, and for the article content to 0.6. The smoothing parameter for Dirichlet smoothing is set to $\mu = 2500$, except otherwise stated. For articles without article intent models, we set $\xi = 0.5$. We report on standard IR measures: precision at 5 (P@5), mean reciprocal rank (MRR), mean average precision (MAP), and r-precision (Rprec). Statistical significance is tested using a two-tailed paired t-test and is marked as $^{\blacktriangle}$ (or $^{\blacktriangledown}$) for significant differences for $\alpha = .01$, or $^{\vartriangle}$ (and $^{\triangledown}$) for $\alpha = .05$.

8. RESULTS AND ANALYSIS

In this section we report on the results of our three experiments: (a) query models and temporal bias in retrieval, (b) weighting schemes for generating query mixture models, and (c) retrieval on article intent model index.

Query modeling. In our first experiment, we test our three query modeling methods we described in Section 5.3: (a) the incoming article (ART), (b) the article intent models (AIM), and (c) their combination ($\mathrm{ART} + \mathrm{AIM}$). These models are tested on two retrieval models, one with, and one without temporal bias. Models on the retrieval model with temporal bias are denoted with a superscript T. Our baseline is set to the method that uses only the incoming article (AIM, and $\mathrm{ART}^{\mathrm{T}}$).

In Table 3 we report on the performance of these systems with (top-half) and without (bottom-half) temporal bias in the retrieval process. In the retrieval setting without temporal bias, the baseline proves strong, and outperforms both AIM, and $\mathrm{ART} + \mathrm{AIM}$. In the retrieval setting with temporal bias the picture changes. $\mathrm{ART}^{\mathrm{T}}$ outperforms AIM in MAP, MRR, and P@5. $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$, the combination of incoming article, and the most similar article intent model, yields the best run, and outperforms the baseline in all metrics, statistically significantly so.

We explain the lower performance of AIM, and $\mathrm{AIM}^{\mathrm{T}}$ (using only article intent models) by the fact that both models are dependent on the similarity of the incoming article to the article intent model. This dependency results in many instances to model the incoming user query–article pair with article intent models that are topically far away from the input pair. This sensitivity is smoothed out successfully in $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ where content from the input pair reduces the potential topical drift from the article intent model.

Weighting. In our second experiment we compare the effect of the weighting methods we describe in Section 5.4. We set the baseline to the best run so far, $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$, which uses uniform weights. The retrieval method is temporally biased.

In Table 4 we report on results for our five weighting schemes. $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ marks the best performance, outperforming other weighting methods with statistical signifi-

[2] http://sourceforge.net/projects/supersensetag/

[3] http://www.timeml.org/site/tarsqi/

Table 2: Retrieval models we consider.

		Input model				
Model	**Temp.Prior**	**Article**	**# AIM**	**Enhanced**	**Weighting scheme**	**Eq.**
Models retrieve articles						
ART	—	✓	—	No	—	(2)
ART$^{\mathrm{T}}$	✓	✓	—	No	—	(2)
AIM	—	—	1	No	—	(3)
AIM$^{\mathrm{T}}$	✓	—	1	No	—	(3)
ART + AIM	—	✓	1	No	Merge	(4)
ART + AIM$^{\mathrm{T}}$	✓	✓	1	No	Merge	(4)
ART + AIM$^{\mathrm{T}}_{\mathrm{P}}$	✓	✓	1	No	Pairwise	(4)
ART + AIM$^{\mathrm{T}}_{\mathrm{PR}}$	✓	✓	1	No	Pairwise-R	(4)
ART + AIM$^{\mathrm{T}}_{\mathrm{A}}$	✓	✓	N	No	Average	(4)
ART + AIM$^{\mathrm{T}}_{\mathrm{M}}$	✓	✓	N	No	Median	(4)
Models retrieve AIMs						
AIM − AIM	—	—	1	No	—	
AIM − AIM$_{\mathrm{e}}$	—	—	1	Yes	—	

Table 3: Retrieval performance for three query modeling methods using: a) only the incoming article, b) only article intent models, c) a combination of the two, with and without temporal bias in retrieval. Boldface indicates best performance in the respective metric. Statistical significance tested against ART.

Run	**Rel.Ret.**	**MAP**	**RPrec**	**MRR**	**P@5**
Without temporal bias					
ART	239	0.2775	0.1916	0.2871	0.0889
AIM	200	0.2349$^{\triangledown}$	0.1778	0.2546	0.0779$^{\triangledown}$
ART + AIM	234	0.2619	0.1832	0.2800	0.0889
With temporal bias					
ART$^{\mathrm{T}}$	253	0.3103	0.2216	0.3230	0.1009
AIM$^{\mathrm{T}}$	193	0.2450$^{\blacktriangledown}$	0.1790$^{\triangledown}$	0.2620$^{\blacktriangledown}$	0.0797$^{\blacktriangledown}$
ART + AIM$^{\mathrm{T}}$	261	**0.3385**$^{\vartriangle}$	**0.2561**$^{\vartriangle}$	**0.3568**$^{\vartriangle}$	**0.1083**$^{\vartriangle}$

Table 4: Retrieval performance for five weighting schemes for creating input article–article intent mixture models. Statistical significance tested against ART + AIM$^{\mathrm{T}}$.

Run	**Rel.Ret.**	**MAP**	**RPrec**	**MRR**	**P@5**
ART + AIM$^{\mathrm{T}}$	261	0.3385	0.2561	0.3568	0.1083
ART + AIM$^{\mathrm{T}}_{\mathrm{P}}$	252	0.3159	0.2289	0.3284$^{\triangledown}$	0.1037
ART + AIM$^{\mathrm{T}}_{\mathrm{PR}}$	252	0.3110$^{\triangledown}$	0.2254	0.3238$^{\blacktriangledown}$	0.1014$^{\triangledown}$
ART + AIM$^{\mathrm{T}}_{\mathrm{A}}$	253	0.3116$^{\triangledown}$	0.2289	0.3252$^{\triangledown}$	0.1009$^{\triangledown}$
ART + AIM$^{\mathrm{T}}_{\mathrm{M}}$	249	0.3104$^{\triangledown}$	0.2289	0.3248$^{\blacktriangledown}$	0.1000$^{\blacktriangledown}$

cant differences in most metrics. Among the rest of weighting schemes, performance hovers at similar levels. We believe this is a indication that the semantic similarity between the incoming article, and the article intent models may not be as discriminative as we hypothesized for assigning weights.

Retrieval in intent space. In our third experiment, we look at methods that retrieve article intent models instead of articles. We use Eq. (3) for query modeling. Then, we issue the query to an index of article intent models. The retrieved article models are mapped back to articles. We consider two methods for performing the mapping. AIM − AIM maps a retrieved article intent model to the most similar article in the dataset, and AIM − AIM$_{\mathrm{e}}$ maps a retrieved article intent model to the I most similar articles.

Table 5: Retrieval performance for two systems retrieving article intent models, and then mapping them to articles.

Run	**MAP**	**RPrec**	**MRR**	**P@5**
AIM − AIM	0.1664	0.1025	0.1821	0.0659
AIM − AIM$_{\mathrm{e}}$	0.1431	0.0895	0.1608	0.0512

In Table 5 we report on the performance of the two methods. The results are not directly comparable to those reported for retrieving articles because we are using a different index (an article intent model index), we observe a decrease in performance compared to the methods that directly retrieve articles. We foresee two reasons for the drop in performance. First, moving from an input article to an article intent model is an error prone process, because of the topical noise. This issue was also present in our first experiment when we used only article intent model for query modeling. Then, when we move back from the retrieved intent models to articles, additional noise is added multiplying the negative effects in retrieval effectiveness.

In sum, our experiments demonstrate the utility of our query models to capture user intent for predicting articles that a user will visit next. The most successful strategy is to use information from both the input query and article, and article intent models for query modeling. For mixing the two sources, uniform weighting proves the most effective. Performance is further improved with the user of temporally aware retrieval models. In the next section we further discuss our findings.

9. DISCUSSION

To better understand the performance of our methods, we take a closer look at the results, and we perform an analysis in the following directions: (a) temporal modeling, (b) the number of article intent models we consider, and (c) parameter optimization.

Temporal modeling. In our temporal aware retrieval models, we use the Cauchy distribution for modeling the bias of

Table 6: Retrieval performance for three temporal models using: (a) Cauchy distribution , (b) a block function, and (c) Laplace distribution. Statistical significance tested against $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$.

Model	Rel.Ret.	MAP	RPrec	MRR	P@5
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$	261	0.3385	0.2561	0.3568	0.1083
Block	279	0.3214	0.2266$^{\triangledown}$	0.3398	0.1046
Laplace	251	0.3299$^{\triangledown}$	0.2527	0.3485$^{\triangledown}$	0.1041$^{\blacktriangledown}$

users towards recent news articles. We try to fit different temporal models on the distribution shown in Fig. 3. In particular, we look at a block function, and at Laplace distribution. From the shape of the distribution in Fig. 3 we derive the block function:

$$F(x) = \begin{cases} e^{-x+2}, & x > 2, \\ e^{x}, & 2 \leq x \leq 2, \\ e^{x-2}, & x < 2. \end{cases}$$

The Laplace distribution is defined as:

$$F(x) = \frac{1}{2b} \begin{cases} e^{-\frac{\mu - x}{b}}, & x < \mu, \\ e^{-\frac{x-\mu}{b}}, & x \geq \mu. \end{cases}$$

with $\mu = 0, b = 1$. We test the potential of these models on our best run, $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$, replacing the Cauchy prior with a prior from the block function, and the Laplace distribution, respectively. The Cauchy prior marks the best performance among the temporal models. Comparing the Laplace distribution to the block function, the Laplace distribution recalls less documents, with higher precision (Rprec). The block function shows the opposite behavior; it shows the highest recall among all methods in expense of precision (see Table 6).

Number of article intent models. In our experiments for query mixture models we used one article intent model, the most similar to the input article. Here, we explore the effect on retrieval effectiveness by increasing the number of article intent models we consider.

In Table 7 we list the results from combining one, up to four article intent models with the input article. On average, increasing the number of article intent models leads to a decrease in performance for all methods. $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ achieves the best performance across the board for $N = 1$. Each method peaks at different number of article intent models; $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{A}}$ peaks at $N = 3$, and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$ at $N = 2$. The differences in performance at various N, however, for the later two models are small.

The performance of $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ decreases quickly as N increases. A possible explanation can be due to the uniform weights assigned to the input article and to the article intent models. Uniform weights allow article intent models topically far away from the input article to be weighted equally, multiplying the effects of topical drift. The weighting schemes of $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{A}}$ and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$ manage count for this effect, and show relatively stable performance for all N.

Parameter optimization. We explore the effect of the language model smoothing parameter on retrieval effectiveness. In our particular setting, the query models are much longer compared to traditional web search queries because they contain content of news articles, and for query mixture models, contain content from several news articles. We perform a parameter sweep on the Dirichlet priors smoothing parameter μ for two runs, ART, and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$. Fig. 4 illustrates the retrieval performance against μ. The differences in performance for different values are small. We believe this is due to the large size of the query, which lessens the smoothing effects.

Table 7: MAP scores for three weighting schemes for combining one to four article intent models with the incoming article. Boldface indicates best performance for the respective model.

	# Article intent models			
Model	1	2	3	4
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$	**0.3385**	0.2276	0.1878	0.1764
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{A}}$	0.3116	0.3106	**0.3141**	0.3037
$\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}_{\mathrm{M}}$	0.3104	**0.3107**	0.3085	0.2990

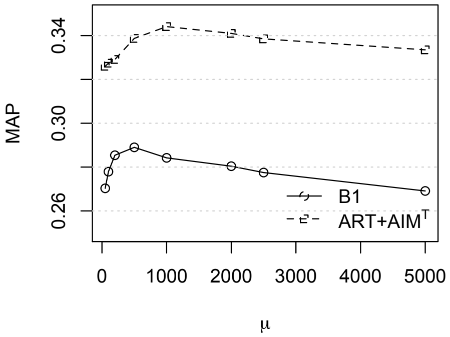

Figure 4: Retrieval effectiveness in MAP for the runs ART, and $\mathrm{ART} + \mathrm{AIM}^{\mathrm{T}}$ over a range of values of smoothing parameter μ.

10. CONCLUSIONS AND OUTLOOK

In this work we have introduced the task of temporal context discovery: Given a query from a user, and the first document they visit, the system aims to predict documents that are likely for the user to visit next. The task takes place in near real-time and systems need to suggest documents not necessarily seen before. The system tries to capture the user browsing intent, and to take into account the change in intent after the user visits the first document.

We focused on an instantiation of the task, and in particular on the news domain. We approached the task as a retrieval problem, and developed query modeling methods that aim to capture user intent. For this purpose, we introduced the *article intent models*, which are trained on the content of user queries and news articles that users have had visited, extracted from user trails in query logs. We presented methods for modeling article intent models with the input query and new article, and several weighting schemes for generating query mixture models. Our experiments demonstrate the utility of our methods for predicting news articles that are visited from users.

In future work, we envisage to enhance our query modeling methods with more elaborate term selection and weighting schemes. Also, we plan extending our query models for incremental updating so we are able to make suggestions given parts of a user trail. Finally, we would like to validate the models presented here with a user-based study, to determine whether the effect of the suggestions produce any behavioral difference in human readers. We believe this line of work is useful to online news agents for increasing the user engagement of their web presence.

Acknowledgments. The authors would like to thank the anonymous reviewers for their comments.

References

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*, pages 19–26, New York, NY, USA, 2006. ACM.

[2] A. Ashkan, C. L. Clarke, E. Agichtein, and Q. Guo. Classifying and characterizing query intent. In *ECIR '09*, pages 578–586, Berlin, Heidelberg, 2009. Springer-Verlag.

[3] S. Attfield, G. Kazai, M. Lalmas, and B. Piwowarski. Towards a science of user engagement (position paper). In *WSDM Workshop on User Modeling for Web Applications*, February 2011.

[4] M. Bilenko and R. W. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *WWW '08*, pages 51–60, New York, NY, USA, 2008. ACM.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *CIKM '08*, pages 609–618, New York, NY, USA, 2008. ACM.

[6] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36:3–10, September 2002.

[7] L. Calderon-Benavides, C. Gonzalez-Caro, and R. Baeza-Yates. Towards a deeper understanding of the user?s query intent. *Search*, pages 1–4, 2010.

[8] O. Chapelle, S. Ji, C. Liao, E. Velipasaoglu, L. Lai, and S.-L. Wu. Intent-based diversification of web search results: metrics and algorithms. *Inf. Retr.*, 14 (6):572–592, dec 2011.

[9] M. Gamon, S. Basu, D. Belenko, D. Fisher, M. Hurst, and A. Konig. Blews: Using blogs to provide context for news articles. *Association for the Advancement of Artificial Intelligence*, 2008.

[10] C. N. González-Caro and R. A. Baeza-Yates. A multi-faceted approach to query intent classification. In R. Grossi, F. Sebastiani, and F. Silvestri, editors, *SPIRE*, volume 7024 of *Lecture Notes in Computer Science*, pages 368–379. Springer, 2011.

[11] J. Guo, X. Cheng, G. Xu, and X. Zhu. Intent-aware query similarity. In *CIKM '11*, pages 259–268, New York, NY, USA, 2011. ACM.

[12] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, New York, NY, USA, 2002. ACM.

[13] N. Kanhabua, R. Blanco, and M. Matthews. Ranking related news predictions. In *SIGIR '11*, pages 755–764, New York, NY, USA, 2011. ACM.

[14] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *SIGIR '04*, pages 377–384, New York, NY, USA, 2004. ACM.

[15] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW '05*, pages 391–400, New York, NY, USA, 2005. ACM.

[16] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW '10*, pages 661–670, New York, NY, USA, 2010. ACM.

[17] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, 40:735–750, September 2004.

[18] D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel. Similarity measures for tracking information flow. In *CIKM '05*, pages 517–524, New York, NY, USA, 2005. ACM.

[19] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07*, volume 7, pages 233–242, 2007.

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

[21] R. L. Santos, C. Macdonald, and I. Ounis. Intent-aware search result diversification. In *SIGIR '11*, pages 595–604, New York, NY, USA, 2011. ACM.

[22] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33:6–12, September 1999.

[23] A. Singla, R. White, and J. Huang. Studying trailfinding algorithms for enhanced web search. In *SIGIR '10*, pages 443–450, New York, NY, USA, 2010. ACM.

[24] M. Tsagkias, M. de Rijke, and W. Weerkamp. Hypergeometric language models for republished article finding. In *SIGIR '11*, pages 485–494, New York, NY, USA, 2011. ACM.

[25] M. Tsagkias, M. de Rijke, and W. Weerkamp. Linking online news and social media. In *WSDM '11*, pages 565–574, New York, NY, USA, 2011. ACM.

[26] R. W. White, P. Bailey, and L. Chen. Predicting user interests from contextual information. In *SIGIR '09*, pages 363–370, New York, NY, USA, 2009. ACM.

[27] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22:179–214, April 2004.