Ranking of Daily Deals with Concept Expansion

Roi Blanco, Michael Matthews, Peter Mika

Yahoo Labs
Barcelona, Spain

Abstract

Daily deals have emerged in the last three years as a successful form of online advertising. The downside of this success is that users are increasingly overloaded by the many thousands of deals offered each day by dozens of deal providers and aggregators. The challenge is thus offering the right deals to the right users i.e., the relevance ranking of deals. This is the problem we address in our paper. Exploiting the characteristics of deals data, we propose a combination of a term- and a concept-based retrieval model that closes the semantic gap between queries and documents expanding both of them with category information. The method consistently outperforms state-of-the-art methods based on term-matching alone and existing approaches for ad classification and ranking.

Keywords: Deals ranking, query expansion, text classification, semantic search

1. Introduction

Daily deals have become a popular advertising model in recent years. The first and to date largest company to promote a business model based on daily deals has been Groupon. Founded in 2008, Groupon has been the first company ever to reach a revenue of over 500 million dollars in its first three years of existence and in 2011 it completed the largest IPO in Internet history since Google's initial offering in 2004.[1]

Email addresses: roi@yahoo-inc.com (Roi Blanco), mikemat@yahoo-inc.com (Michael Matthews), pmika@yahoo-inc.com (Peter Mika)

[1] http://www.reuters.com/article/2011/11/04/us-groupon-idUSTRE7A352020111104

Soon adopted by a number of competitors, Groupon's business model is based on a novel form of advertising offered to businesses. In this model, advertisers offer significant discounts on products or services, but impose certain limitations. The offers presented are time-bound and consumers often need to make a decision within the day, hence the name daily deals. Further, the offer only becomes valid if a minimum number of buyers agree to purchase (also known as an assurance contract). Other limitations may exist e.g., the offers may be limited to particular geographies or store locations, which allows merchants to target particular markets where they face competition. Daily deals are a form of direct-response marketing, in that the results are directly measurable. From the consumers perspective, the limitations of daily deals increase the thrill of participating in a deal and favor impulse buying. This also means that daily deals are typically offered for products and services that a consumer is willing to agree to buy instantly. As in the case of coupons, daily deals are known to attract price-conscious customers.

In the original business model, the *deal provider* such as Groupon finds merchants who are willing to advertise, and helps them to formulate the creative (deal text and image). The deal provider is also responsible for finding the customers and for this it maintains a mailing list of subscribers, along with a website and mobile application where visitors can search and browse the deals. The right targeting i.e., matching the subscribers to the deals that might interest them is critical in that the advertiser does not pay in advance, but provides a revenue share; Groupon and competitive deal providers typically take about 50% of the revenue from the deal. Lately, a second business model is emerging as well, in which a *deal aggregator* receives data from multiple deal providers or other aggregators and shows the combined set of deals to its users. Deal aggregators work in an affiliate model, forwarding customers to the original deal provider and receiving a fixed price per click or a share of the revenue. General purpose web search engines such as Bing, Google and Yahoo also act as either deal providers, aggregators or both and show deals among their web search results.

The information retrieval problem of ranking deals is critical to the success of all participants. There is a limited space to show deals on both search engines, provider and aggregator websites and the mailing lists, and users are quickly overloaded with offers. Thus deal providers and aggregators need to find the most relevant deals to show for each user. We note that deal providers (but not aggregators) also have the allocation problem of making sure each deal gets enough users for the deals to get activated i.e., that each

deal attracts enough users to at least hit the minimum number of buyers required. In this paper, we restrict ourselves to the problem of relevance ranking.

In the following, we study the problem of ranking deals from the perspective of a deal aggregator, Yahoo Deals which maintains its own website and mailing list, and integrates deals into Yahoo Search, a web search engine. For deal aggregators such as Yahoo Deals, the ranking problem is particularly acute given the (1) larger number of deals and (2) the heterogeneous collection of deals. Figure 1 shows an example of how deal search is integrated into the user experience in web search. When the user is searching for a product, we show relevant products, buying guides as well as ways to save through deals and coupons. Selecting the top-4 deals for this display is one of the implementations of our work.

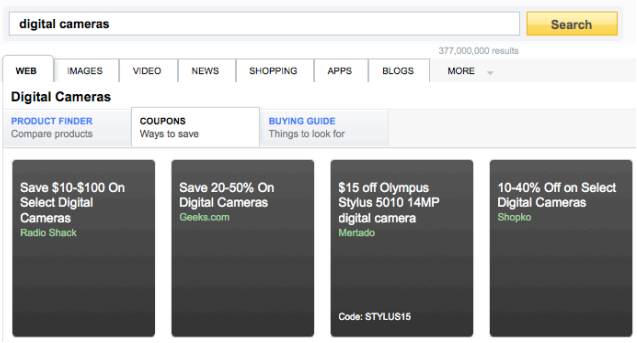


Figure 1: Integration of deals retrieval in a web search engine

The problem of ranking daily deals as a response to a user query can be regarded as similar to that of ranking creative ads or Tweets. In fact some particular aspects of the solution we provide here could potentially be helpful in any situation in which term-sparsity is a severe issue, such as sentence, tweet or ad ranking.

There are, however, some differences that are particular for deals. First of all, deals contain metadata that hints important signals for ranking, provenance, dates in which the deal is active, location, merchant among others. Secondly, deals are textual units that should be related to a particular business category, and as such there is a precise correspondence to which deals could be surfaced to particular information needs. Deals are aggregated in a final end system through a plethora of different sources; metadata, and

categorization must be reconciled first into a classification taxonomy that is well understood by the search engine. In addition, these categories might be not completely reliable so a prior classification step is mandatory. Secondly, deals have longer text than typical ads, which contain a few keywords for triggering. Language-wise, deals are written with a narrow set of words, specific per business domain, unlike text that is found in social media or micro-blogging sites [20].

To our knowledge, ours is the first paper to study the problem of ranking deals, and therefore we begin by characterizing the data and the most common retrieval tasks. In this work, we study in detail the task of ranking deals for an ad-hoc query. To solve this problem, we propose a novel retrieval model that combines text-based retrieval with concept-based retrieval, in particular taxonomy-based matching. We will show that this model is particularly adept in addressing the deals ranking problem because semantic matching effectively deals with the sparsity of deal text and the resulting semantic gaps between the query intent and document content. The model also exploits some of the metadata associated with deals, in particular the profile of the merchant who is offering the deal. We evaluate the effectiveness of the method on a multi-day collection of data from Yahoo Deals by comparing it to BM25F, a query-expansion method from the literature and an existing approach for ranking creatives. We find that the best performing approach introduced in this paper is able to display roughly twice the amount of relevant deals as a state-of-the-art keyword retrieval model, outperforming existing approaches by as much as 40% in NDCG.

The paper is organized as follows. Section 2 reviews related work, Section 3 presents the retrieval tasks we address, the data and models and Section 5 contains our experimental results. The paper finishes with a conclusions Section.

2. Related work

Byers et al. provided a unique analysis of daily deals sites from a microeconomic perspective, based on data the authors have collected by crawling Groupon and LivingSocial periodically in the first half of 2011 [8]. Using regression analysis, they described a model for predicting deal size (the number of deals purchased) based on the price, the deal threshold, duration and deal attributes such as whether the deal was featured or not, and whether the

inventory was limited or not, the category of the deal, the location and the day of the week in which it was posted. In an extended version of the model, they also added social features such as the star-rating of the merchant and the number of Facebook likes that the deal receives. This model achieved a reasonable predictive power, and can be used to predict the eventual success of the deal based on the deal attributes alone. The model can also incorporate early sales data to improve prediction e.g., to predict eventual sales at the end of the day based on data from sales in the first hours. The analysis of the regression coefficients also provides insight into the factors that make a deal successful. This analysis is particular useful for deal publishers who have a control over some of these parameters e.g., which deals to feature and when to schedule them, and in fact the authors find evidence of deal programming (e.g., that Groupon selects featured deals belonging to different categories on consecutive days or that they prefer to launch deals on Friday that span the entire weekend).

The key difference to our work is that this analysis reflects the perspective of the deal provider, whose goal is to maximize deal size for each individual deal and thereby satisfying their primary customers, the merchants. This is not the same as optimizing relevance from the perspective of each individual user, the primary customer of a deal aggregator. The position of a deal aggregator is also different from that of a deal provider. The aggregator can not influence the deal supply (e.g., change the attributes of a deal or schedule deals) and has to select deals for their users from the collections provided by the deal providers. Similarly, deal aggregators do not have access to much of the data available to the deal provider e.g., early sales data or Facebook Like data associated with the original deal page.

Despite the relevance of the problem to both merchants, consumers and deal sites, to our knowledge ours is the first paper to study the problem of ranking deals. Given the observed characteristics of the deals data in Section 3.2, we propose a concept-based retrieval method to address the semantic gap between query intent and document representation. Previous works have already experimented with a variety of approaches addressing the same need, for example by exploiting the corpus, relevance feedback or external knowledge. Topic modeling approaches such as LSI [11] and pLSI [13] capture implicit concepts by reducing the dimensionality of the term-document matrix and use the discovered hidden or implicit topics as the space in which to match queries and documents. In effect, this approach results in query and document expansion, producing matching even when the

query and document share no terms in common. Wei [29] provides a survey of topic modeling for information retrieval. Methods based on explicit semantics exploit additional knowledge in the form of thesauri (e.g., WordNet) or domain ontologies to perform query and document interpretation and expansion [12, 15, 10]. Many of these approaches are also costly in terms of the human effort that goes into creating domain specific ontologies, concept or entity recognizers. Instead, we rely on a generic classification method that is cheap to deploy and efficient to run.

Concept-based expansion [21] is a variant of query expansion that makes use of a similarity thesaurus to reflect domain knowledge about the particular collection from which it is constructed. In this stream of research, the work of Broder et al. [6] is the most similar to ours in terms of the method, and in that they also find positive effects of using semantic matching in retrieval. Their focus is to classify ads, whereas we target directly the problem of relevance ranking of deals. They use a convex combination of keyword score and taxonomy score and they compute the distance between the query and document categories in the online phase of ranking, which provides poorer performance than our approach of pre-computing all relevant categories. In addition to their work, we investigate the impact of the level of classification and show that multi-level classification adds value to the results. We show that our approach performs better for the task of query-based deals retrieval. Selecting the right keywords is crucial for effective sponsored search; Joshi and Motwani present a method to infer term semantic relationships, which are modelled as a directed graph in order to provide a large number of related keywords [14]. Chen et al. [9] further explored the suggestion of keywords in the context of advertising, which come from a concept hierarchy. They did not, however, evaluate their method in a retrieval scenario.

In further related work, Bennet et al [4] present a simple framework that uses clicks in combination with classification of web pages to derive a class distribution for queries. They use the class distribution to derive features that are inputted to machine learning ranking model. In our set-up we do not make use of any click-through information, given that the domain is very dynamic (deals typically expire after a day or two) and therefore the system would require a high traffic rate per query (or query class) to acquire sufficient signal to learn an effective model. Query expansion has a long history in the IR community [24, 7, 2, 16], with a broad number of different approaches, from Rocchio's seminal paper [24] to relevance models [16]. Most of these methods build an extended representation of the query based on the

occurrences of terms in the top ranked documents of the original query [18]. There exists, however, other more sophisticated approaches which employ the information collected from a user's session to augment the search context [26] or adapt the parameter set-up to balance the feedback influence in retrieval on a per-query basis [17]. In this paper we will use a method derived from Bose-Einstein (Bo1) statistics [1] integrated into the Divergence from Randomness (DFR) framework [2].

We note that many of the above methods have a mixed track record in Information Retrieval. Typically, they show inconsistent results across queries: while results on some queries improve, other queries show worse results. In the practice of running a search service, this is a problem even when the net effect on retrieval performance is positive: users may abandon their search when faced with a single bad experience. Although we lack widely accepted metrics, we will attempt to show that our improvements are consistently positive across queries.

3. Retrieval Challenges

Since to our knowledge ours is the first work to study the problem of relevance ranking in this domain, we describe briefly the various retrieval tasks associated with deals and the data that is available, before we formalize our retrieval model.

3.1. User tasks

As described in Section 1, users primarily interact with deals by browsing specialized deal websites or mobile applications, subscribing to mailing lists or searching in either within the deal sites or general purpose web search engines. Based on this, we can distinguish the following three main retrieval tasks:

1. Deal recommendation. Visitors of daily deal websites and the users of mobile applications may not have an express shopping need. In this case, the retrieval engine recommends deals based on implicit or explicitly defined user interests and context information such as the user's geographic distance from a local business that is offering a deal.
2. Deal routing. Similar to the previous case, subscribers of daily deals mailing lists may not have a specific shopping need, or if they do the deals provider may not be aware of it. Subscribers specify the location

where they live and some providers and aggregators also allow to restrict deals by category e.g., LivingSocial asks subscribers if they are interested in family-focused deals.

3. Deal search. Users with an explicit shopping need may search for deals either on specialized websites or using web search engines that integrate deals among their results.

The last task is different from the first two in that the relevance of the results can be more easily evaluated by comparison to the user's specific information need. In the first two cases, even if the user may have a specific shopping need, this is not explicit. We expect however that many of these users are simply "hunting" for a deal that matches their interests. Deals usage data suggests that indeed some deals are able to convert a large number of users to buyers, such as the well-known case when Amazon offered a $20 gift card for $10.[2] We thus suspect that deals have a generic quality based on the amount of discount offered, the typical discount and the range of products to which the deal applies. This static quality of deals may be exploited in all three tasks. However, we leave the investigation of this factor to future work.

In this paper, we address the third problem. Yahoo is a deal aggregator that has maintained a separate website for deals[3] for the past years. In the current work, we describe how we integrated deals among the results of the Yahoo web search engine. Though web search engines serve many needs, a significant portion of web search queries relate to shopping, where deals may be relevant. An editorial analysis of 12,340 queries sampled from our US query logs collected during the first six months of 2010 showed us that 18.4% of queries have a potential shopping intent. 8.3% of the queries explicitly mention a product or product category, while 5.3% of the queries name a store or brand (the remaining queries are for manuals, reviews, downloads, support etc.) The queries were first classified into broad categories (news, shopping, local, etc.) and further analyzed based on the type of entity mentioned in the query (product/product category, store, brand etc.) The shopping category was defined as queries that seek information on "tangible, physical products that can be purchased over the Internet".

[2] http://latimesblogs.latimes.com/money_co/2011/01/livingsocial-creates-frenzy-by-selling-20-amazon-gift-cards-for-10.html

[3] http://deals.yahoo.com

We can expect that in particular those closer to a purchase decision would benefit from a deal that gives a discount on the product or product category they are looking for. However, most queries in our logs represent the earlier stages of the shopping process. This is in the nature of the task: the shopping process is often described as a funnel where more and more specific queries are issued by fewer and fewer users as many more users are interested in "shopping around" than making actual purchases. It is important to remark that this effect is product- and price-dependent.

3.2. Provider Data

46" Sony Bravia LED Edgelit HDTV $1000 Sony KDL46NX720 BRAVIA 46" 3D LED Backlit HDTV drops from 2099.99 to 1099.99 with free shipping when you enter code UTV52750 at checkout
80% off - $25 Restaurant.com certificates for Only $4 Use Code: ENJOY - Enjoy meals for less than half price and find great restaurants in your area with Restaurant.com! Search over 18,000 restaurants nationwide and easily print certificates at home.
90-Minute Structural Integration Bodywork Session The seven dwarves may whistle while they work, but all those trips to and from the mine have resulted in a variety of namesake ailments: Sneezy, Sleepy, Dopey, Bashful, Grumpy, and Slouchy. It's a good thing they have Doc to refer them to today's deal: $60 gets you a 90-minute Structural Integration bodywork session at Sedona Pointe Integrative Bodywork (a $120 value) – a feel-good antidote to feeling glum and Grimm. *+104 more words*

Table 1: Example deals title and text. The first two examples are coupons. The third example is a local deal.

Deal aggregators such as Yahoo receive data from multiple providers. In our case, we aggregate deals (both local and online deals) and coupons. Coupons do not require advance purchase, but rather contain a code that can be used to get a discount. However, in general both coupons and other deals provide the same benefit to the user. In the following, we will use the term deal for coupons as well, unless otherwise noted.

Deals data is highly structured and exchanged in the form of XML feeds among partners. Since there is no standard data format for deals, we take care of mapping the data from the provider-specific schema into a global

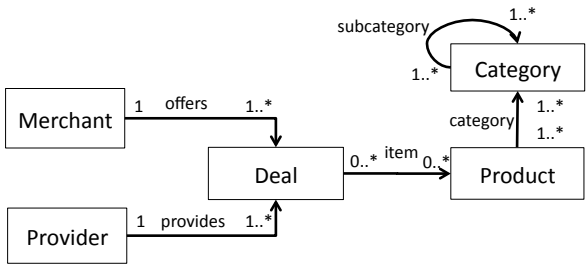


Figure 2: Structure of the deals data

schema. Since this global schema is a super-set of the input schemas, not all deals will have values for all attributes. We also take care of cleaning the data and enriching it, for example coding addresses into geo-locations.

Figure 2 shows an object-oriented view of the deals data from the perspective of the aggregator. In the following, we detail the key attributes with respect to each class. A Deal consists of a short title, a text and optionally an image and additional information such as the target location, the start- and end-date of the offer and other metadata. Although we do not consider this information in our current work, the deal aggregator or the deal provider may collect usage data such as click-through information, the quantity sold up to a given moment or social sharing information such as the number of Facebook likes for the deal.

Arguably, location and start/end dates of a deal are critical bits of information for deals ranking. From a text information retrieval perspective, however, the most salient aspect of deals are the text fields that contain the title and the description of the deal. Table 1 shows a few examples of deal title and text. Here there is a substantial difference between coupons and other types of deals. Coupons are short, on average 6.3 tokens and 34.1 characters in length. For non-coupon deals, descriptions are longer runs of text containing 175.4 tokens and 1286 characters on average. See 5 for a description of our dataset. However, much of this text is creative writing as we can see in the third example.[4]

This means that after removing stylistics, there is still a sparsity of text compared to the potential queries to which a deal might be relevant, in fact in

[4]In fact, many providers such as Groupon employ their own writers who work with the advertiser to produce the text.

many cases a relevant deal may not contain any of the query terms (we refer to the well-known term mismatch sparsity problem). In addition, the short length of the text means that there is a semantic gap between the user intent and the deal text. The problem can be illustrated on the first two examples in Table 1. In the first case, the deal is for a particular product and contains all the specific terms to identify the product e.g., *HDTV* and *LED*. However, it does not contain trivially relevant but much broader query terms such as *television* or *tv*. The second example demonstrates the opposite case. Here the deal talks about *restaurants* in general, but misses more specific terms such as *pizza* or *chinese*. In terms of queries, the first type of semantic gap is more typical. As noted above, there are more users at the beginning of the shopping funnel, where the highly specific (i.e. high idf) terms such as model names and numbers are typically missing. This is because in the early stages of the shopping process users do not have a specific model in mind but rather search by the description of the product they are looking for.

In summary, the questions we aim to answer are the following:

- Can we bridge semantic gaps using concept-based retrieval methods more effectively than what is possible using traditional keyword-based query expansion?
- Is it better to use fine-grained or broad classifications? Is there a way to exploit the advantages of classification at different levels of granularity?
- Can we exploit additional deal specific metadata such as the identify of the seller?

4. Retrieval Method

In the following we describe our solution for deals retrieval, which combines taxonomy-based classification with

The key intuition for our work is that a concept-based retrieval method can successfully address some of the problems described above, in particular the sparsity of the deal text, and the semantic gap between deal text and query terms. Our system utilizes category features assigned using an automated classifier, and combines them with text-based features in a single unified retrieval model. We thus first describe the classification method in Section 4.1 and show how we integrate term-based and concept-based retrieval in Section 4.2.

4.1. Classification model

The deal text may describe or otherwise relate to a particular product or service, a category of products or services, or multiple categories, depending on how specific the deal is. Identifying a particular product referenced in a deal is a difficult task, and requires information extraction (NLP) methods to identify the product mention as well as disambiguation against a product catalog. This will trivially fail for the deals that don't mention a specific product. For this reason, we will focus on identifying the most relevant categories of a deal.

Large providers such as Groupon assign their own manual classification to daily deals. However, there is no shared taxonomy across providers and smaller providers do not classify their deals. For this reason, we will develop a simple automated classifier instead of relying the provided classifications. In future work, we will investigate if we could develop a classifier by mapping provider categories to a shared schema.

We classify deals against an e-commerce taxonomy used for product classification on our website Yahoo Shopping.[5] This taxonomy is three levels deep with 26 categories at the top level (level 1), 275 categories at the middle level (level 2) and 1401 leaf-level (level 3) categories. Although this taxonomy is not available for download, it is very similar in nature to other taxonomies used for navigation in large shopping sites and it has not been specifically developed for daily deals. Figure 3 shows a subset of the taxonomy.

With the taxonomy, we have a total of forty million products classified to the leaf level. We use this data for training a basic classifier. In particular, we index the product descriptions of the first ten thousand products in each category. In other words, the documents in this index correspond to a category of products and the content is composed virtually from the text of the products in that category. We then use the deal text as query to this index and retrieve the top-k categories using the state-of-the-art BM25F retrieval method [25]. Although in the product taxonomy each product is assigned to exactly one category, we will experiment with assigning zero, one or more categories to deals, based on the assumption that multiple categories may be relevant to at least some deals or there may be no relevant categories in case our taxonomy does not cover a product category.

Though not often, it also happens that a deal is very generic, as is the

[5] `shopping.yahoo.com`

case for free shipping type offers. In such situation, the relevance of the deal is determined by the type of products typically offered by the merchant. As an example, a free shipping offer from *1800flowers.com* is relevant to flower purchases even if the words *flower* or *delivery* do not actually appear in the text. Therefore we also implement an improvement (we will reference as $Merchant$) that incorporates a model of the merchants who typically offer a category of products. We create an alternate index where for each category (the documents) we index the merchant name for all products in that category. Whenever the original classification returns no results, we retrieve from this index instead, using the merchant as the query text instead of the deal.

This basic classifier achieves moderate accuracy. The classifier returns the correct category[6] with 51.1% accuracy at the lowest level. When classifying at the middle level the accuracy increases to 64% while at the highest level of 26 categories the accuracy is 79.4%. Note that the classifier could be improved in a number of ways, but we are not primarily interested in classification accuracy. Instead, we will show that even with such moderate accuracy the classification successfully improves retrieval performance. In our experiments, we will also look at the question of what level of classification is most suitable to improving retrieval effectiveness. In particular, a fine-grained but less accurate classifier makes more frequent mistakes but captures the concept of the deal better while a broad but more accurate classifier will make less mistakes, but yield less benefits for deal retrieval.

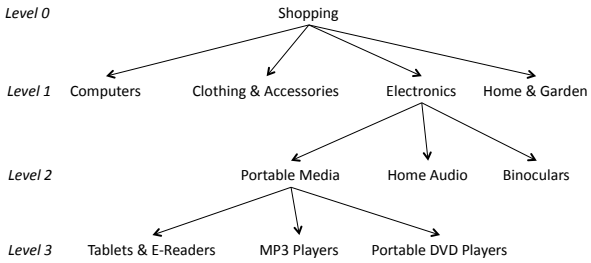


Figure 3: Part of the product taxonomy

[6]Computed using 10-fold cross validation on the whole product catalog.

4.2. Retrieval model

As a solution for integrating term-based and concept-based retrieval we propose a field-based retrieval model that incorporates classification features in the query and document model. In particular, we consider a query as a vector of fields:

$$Q =< \{q\}_i, \{c_1\}_{j_1}, \{c_2\}_{j_2}, \{c_3\}_{j_3} \dots \{c_l\}_{j_l} > \tag{1}$$

consisting of query terms $\{q\}_i$ and $n_j \geq 0$ number of category identifiers $\{c_k\}_j, 0 \leq j \leq n_j$ at each level k of the classification. In our case, the taxonomy has three levels ($l = 3$).

Similarly, we consider documents as

$$D =< \{s_{title}\}_i, \{s_{desc}\}_z, \{c_1\}_{j_1}, \{c_2\}_{j_2}, \{c_3\}_{j_3} \dots \{c_l\}_{j_l} > , \tag{2}$$

where $\{s_{title}\}_i$ and $\{s_{desc}\}_z$ represent the field and description fields. Each one of the fields can be seen as a vector of term frequencies, which in turn are drawn over a common lexicon, even if their statistical properties are different. In practice, however, once a query is scored against a document, the terms that will match the document's category field $\{c_i\}$ will only come from the corresponding query field.

The features that BM25F uses are the field term frequency tf_{si} (number of times term i appears in field s), the field length l_s (number of tokens in the field s) and the field weights v_s, which in our case are computed over the matches of the structured query and document fields. The ranking function does not exploit proximity information or term dependencies.

Using BM25F, a document D is scored against a query Q using a summation over individual scores of query terms $q \in Q$:

$$score(Q, D) = \sum_{q \in Q} w_i^{BM25F} \tag{3}$$

The first of these terms is the BM25F weight of query terms computed by normalizing frequencies using a saturation function:

$$w_i^{BM25F} = \frac{\tilde{tf}_i}{k_1 + \tilde{tf}_i} \cdot w_i^{IDF} , \tag{4}$$

where BM25F aggregates the weighted term frequencies over all the fields S, normalizing them using B_s as

$$\tilde{tf}_i = \sum_{s=1}^{S} v_s \frac{tf_{si}}{B_s} \ , \tag{5}$$

where

$$B_s = \left((1 - b_s) + b_s \cdot \frac{l_s}{avl_s} \right) \ , \tag{6}$$

where avl_s is the average length of field s and b_s is a tunable parameter ($0 \leq b_s \leq 1$) that controls the amount of normalization. where k_1 is a parameter and w_i^{IDF} is the *inverse document frequency* of term i, calculated as $\log\left(\frac{D-n_i+0.5}{n_i+0.5}\right)$ (n_i is the number of documents i occurs in).

5. Evaluation

For evaluation, we collected a set of 4487 deals from Yahoo Deals during the first 10 days of March, 2011. The number of deals in the production system vary over time as deals expire –typically after a day or three days–, and new deals are added dynamically. This collection is a uniform random sample and contains a similar number of deals that Yahoo Deals would typically offer at any time.[7] This is, the experiments used a snapshot of the contents of the Yahoo Deals database that represents its typical state.[8]

Deals contained different textual fields, such as title and description, along with metadata associated with the actual offering, expiration and publication date, whether it offered a discount, merchant identifier, and several urls identifying the provider and the "deal site" landing page.

In preprocessing, we lower-cased all text, split on whitespace and punctuation. We have not applied stemming. We removed numerical expressions such as dollar amounts, percentages and coupon codes and also removed stop-words from a list of 335 terms that we collected from the terms most commonly occurring in our collection of 4487 deals. This list contains both common English terms as well as frequent deal-related terms.

[7]For comparison, 31,646 daily deals are published in a month across all of the 600+ deals websites monitored by Yipit, see `http://yipit.com/data`, August 2011.

[8]As Yahoo is not the original owner but a licensee of deals data, the particular dataset used in this can not be shared publicly.

We first randomly selected a set of 157 queries from the list of queries that triggered the shopping direct display in Yahoo Search web search results.[9] We collected our evaluation queries this way to make sure that they contain a shopping intent, and we do not spend unnecessary resources in evaluation. We then applied the same processing to our queries as to the deal text and further pruned the list of queries by removing those with zero results, resulting in 74 queries.

We classified deals using the classifier introduced in Section 4.1. For queries, we used a slightly different method: we queried the index of products in our product catalog using BM25F as the ranking method and collected the categories of the top five products. In each experiment, we assigned three categories per class-level to the query.

After pooling different deals rankings according to different retrieval models in Section 4.2, we assessed the relevance of each query and result pair by collecting relevance judgments from subject matter experts who are employees of our company. We asked the experts to judge each query by assigning one of four grades (Excellent, Good, Fair or Bad) to each query-result pair. In particular, we asked the judges to mark as Excellent those deals that could be directly useful to the user in getting a discount on the product or product category they are looking for. We used the 'Good' judgment for deals that are potentially relevant e.g., in the case where the discount is for a comparable product. We used Fair in the case where the deal was only slightly relevant e.g., offering a discount on a complementary product such as a deal for *curtains* when the user is looking for *blinds*. All other deals were marked as Bad.

We judged the actual, not the perceived relevance of the deals i.e., we asked five judges to click-through to the landing page at the deal provider. In addition, we provided them with a link to the merchant's home page, and asked them to consult the merchant's catalog when necessary. For example, in case of free shipping type offers, the judges searched the merchant's site to find out if the merchant would be selling the type of product the query is looking for. In total, we collected 12,976 relevance judgments, or on average about 188 judgments per query. Approximately 75% of the relevance assess-

[9]In our operational setting, daily deals are shown as part of the shopping module in Yahoo Search and therefore only queries that trigger this module will invoke the deals retrieval backend. The exact mechanism of triggering this module is outside of the scope of the current work.

ments were labeled as *Bad* by the editors, 14% were Fair, 5% were Good and only 1% Excellent. In the following, for computing the NDCG measure we assign gains of 7, 3, 1 and 0 for Excellent, Good, Fair and Bad, respectively. We performed double assessments on 20 queries in order to check the assessors' agreement obtaining an average Cohen's Kappa of 0.65. Looking at agreement rates in other relevance judgment settings (0.49 on 40 topics at TREC 2006 Legal Track [3], 0.34 on 100 documents for opinion detection [19], 0.55 on sentence relevance at TREC 2004 Novelty Track [27]) we can see that the task could be evaluated objectively based on the provided criteria. We also note that one can have reliable ranking of systems even with lower agreements (precision numbers fluctuate, but the ranking remains the same [5]).

We implemented three baseline methods. The first baseline is the BM25F retrieval method [23, 22] where the fields are the title and description of the deal; this method solely employs term matches and ignores all category information. In addition, we implemented a query-expansion method based on pseudo-relevance feedback. Similar to our methods, query expansion is intended to deal with the sparsity of deal text by adding terms that were missing from the original query, but appear in (expectedly) relevant deals. We use the Bose-Einstein based expansion method (Bo1) of Amati [1], which is in turn a weighted adaptation of tf-idf scoring. In detail, the method operates in two rounds: first we issue a plain BM25F query and secondly, we select t terms out of the k top documents by ranking them according to Eq 7:

$$tf_k \log(\frac{1+P_n}{P_n}) + \log(1+P_n) \ , \tag{7}$$

where tf_k is the term frequency of the term among the top-k ranked documents, and P_n is the ratio between the frequency of the term in the collection and the number of documents. The final weight of the term in the query is also given by Equation 7. We tuned this method extensively by trying all combinations of $1 \leq k \leq 30$ and $1 \leq t \leq 50$. We further experimented with a number of other alternatives, such as plain tf-idf term selection [18], but found that this variant worked best.

Finally, we further compared the performance with Broder et al's [6] method. Essentially, this method ranks ads for a given webpage, employing a convex combination of keyword and category information, weighting the category matches using information on the least common ancestor in the

taxonomy. In our case, the deal plays the role of the ad and the query the role of the webpage. In order to allow fair comparisons, we employed the same taxonomy and classifier to assign categories to deals and queries.

The model that employs the augmented category information, as described in Section 4.2, has two parameters per field, the field weight w and the term frequency normalization parameter b. However, in order to reduce complexity, we decided to use a single b value for all the fields. As mentioned before, the concept-based model uses three different category fields, one per level in the taxonomy, which we refer to as Cat_1, Cat_2, Cat_3 for the top, middle and lower levels respectively. The large number of values for the free parameters in the model precluded us from performing an exhaustive search over the whole parameter space, so we turned to a heuristic learning algorithm to direct the search for optimal parameter values [28]. We selected the best parameter configuration in each experiment by optimizing for MAP using the multidimensional optimization method described by Robertson and Zaragoza [22].

Table 2 and Table 3 summarize the retrieval performance of our method compared to the baselines using P@K, MRR, MAP and NDCG metrics. We tuned the parameters of our baselines to give optimal performance. We report the maximum performance attainable in Table 2 and perform two-fold cross-validation to obtain the test results in Table 3. $BM25F$ and $BM25F + QE$ stand for the baseline without and with query expansion. (Two-fold cross-validation means that we used half of our assessed dataset for training, and half for testing.) $Cat_{1,2,3}$ uses only information from the category fields and not from the textual fields (equivalent to setting the title and description weights to zero), and $BM25F + Cat_x$ uses the textual fields and the corresponding category exclusively. Finally, $BM25F + Cat_{1,2,3}$ uses the information from all the fields available, and $BM25F + Cat_{1,2,3} + Merchant$ uses the fields with the enhanced merchant information, as described in Section 3.2.

In both Tables we show the contribution of individual features by building the best performing model incrementally. In all cases we determined statistical significance over all the baselines using the paired t-test with significance levels at $p < 0.01$. We also note that the cross-validated performance numbers are close to the maximum ones, which indicates that the optimal parameters are stable between folds.

The results highlight the effectiveness of our mixed concept- and keyword-based retrieval method. First, query expansion did not provide significant improvements over BM25F despite the intuition that it should improve re-

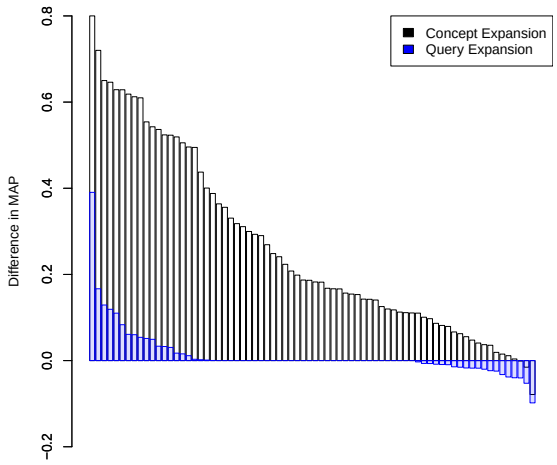


Figure 4: Retrieval performance across queries

trieval in low recall scenarios. On the other hand, using categories alone and ignoring keywords (i.e., matching the categories of the query with categories of the results) outperformed keyword-based retrieval. This is remarkable from a semantic perspective: although our classification is fine grained, it certainly does not capture as much detail as the text. It is also interesting from an efficiency perspective, because classification effectively reduces the deal text to a handful of labels i.e., it is a much more compact representation of the data.

We were also successful in combining text-based and category-based features. Our middle-level categorization gave better improvement on its own than the higher or lower level categorization, which suggests that indeed there is a trade-off between too broad and too narrow classification. We could also effectively combine the three levels of categorization by choosing appropriate weights, which is in line with our expectation that the three levels contribute to relevance in complementary ways, likely by addressing different types of semantic gaps between query and document. Our merchant-based improvement to classification also contributed positively to our models.

We further investigated whether the categories present in the topmost retrieved deals can be used for expanding the original keyword query, in a

similar fashion to the *BM25F + QE* runs. We experimented using the three different categories from the top-most documents for expansion, weighting the category fields and selecting the best categories using Eq 7.

Cross-validated results (MAP) are 0.19, 0.25 and 0.23 respectively for deals-based category expansion using Cat 1, 2 and 3. This category weighting method was able to improve on pure keyword matching and query-based expansion ($BM25F+QE$) although it underperformed with respect to using the product catalog to classify the query. This is due to the fact that the product catalog contains a larger set of items, and therefore it is more likely for the original retrieval pass to find relevant articles (i.e. categories) for a given query just by using keyword matching, and therefore the categories used to expand (or enrich) the query are more accurate.

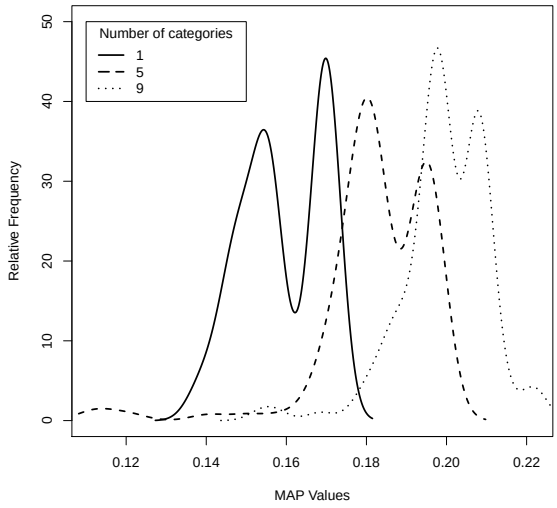


Figure 5: Distribution of MAP values across different number of categories. The plot is a smoothed representation of the performance achieved when varying the different parameters of the model.

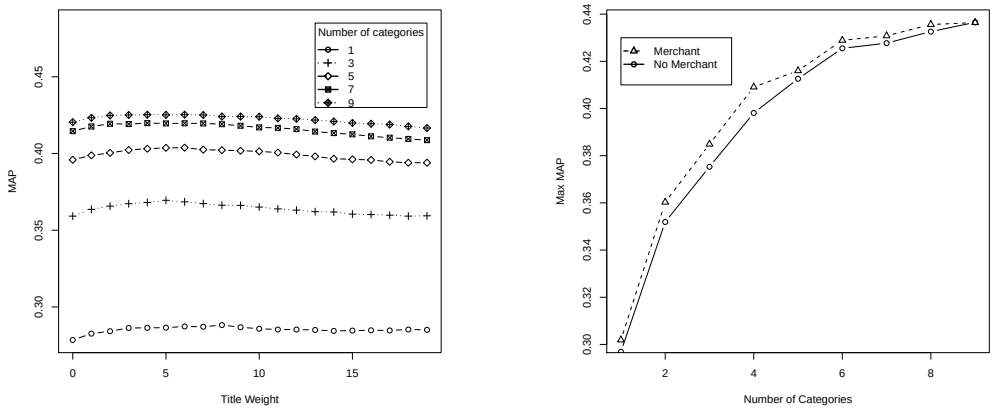
Figure 4 shows the performance improvement per query, ordered by best to worst. It provides evidence that the improvements are consistent across queries. This is key in our view because concept-based retrieval methods have been known to bring improvements on some queries, but worse performance on others. We can see this effect on the results of the QE method, which

Table 2: Maximum effectiveness of each method when using all queries; † indicates statistical improvement over both BM25F and BM25F+ QE using the paired t-test with $p < 0.01$.

Method	P@1	P@3	P@10	MRR	NDCG	MAP
$BM25F$	0.64	0.46	0.30	0.67	0.17	0.13
$BM25F + QE$	0.65	0.49	0.33	0.67	0.16	0.18
Broder et al. [6]	0.54	0.49	0.43	0.65	0.18	0.30
$Cat_{1,2,3}$	0.69	0.63†	0.52	0.74†	0.14	0.32†
$BM25F + Cat_1$	0.73	0.62†	0.47	0.79†	0.17	0.29†
$BM25F + Cat_2$	0.80†	0.66†	0.59	0.86†	0.20	0.38†
$BM25F + Cat_3$	0.76†	0.63†	0.45†	0.80†	0.18	0.26†
$BM25F + Cat_{1,2,3}$	0.80†	0.75†	**0.61**†	0.86†	0.22†	0.43†
$BM25F + Cat_{1,2,3} + Merchant$	**0.84**†	**0.77**†	**0.61**†	**0.88**†	**0.24**†	**0.44**†

Table 3: Two-fold cross-validated effectiveness of each method when using all queries. † indicates statistical improvement over both BM25F and BM25F+ QE using the paired t-test with $p < 0.01$.

Method	P@1	P@3	P@10	MRR	NDCG	MAP
$BM25F$	0.63	0.44	0.29	0.64	0.17	0.12
$BM25F + QE$	0.64	0.48	0.32	0.67	0.16	0.17
Broder et al. [6]	0.52	0.47	0.42	0.62	0.17	0.28
$Cat_{1,2,3}$	0.66	0.57†	0.44	0.74†	0.14	0.26†
$BM25F + Cat_1$	0.69	0.62†	0.47	0.77†	0.17	0.29†
$BM25F + Cat_2$	0.77	0.70	0.57†	0.81†	0.18	0.36†
$BM25F + Cat_3$	0.64	0.60†	0.44†	0.78†	0.18	0.26†
$BM25F + Cat_{1,2,3}$	0.76†	0.75†	**0.61**†	0.82†	0.20†	0.41†
$BM25F + Cat_{1,2,3} + Merchant$	**0.78**†	**0.74**†	**0.61**†	**0.84**†	**0.22**†	**0.42**†



(a) Effect on MAP of varying the title weight

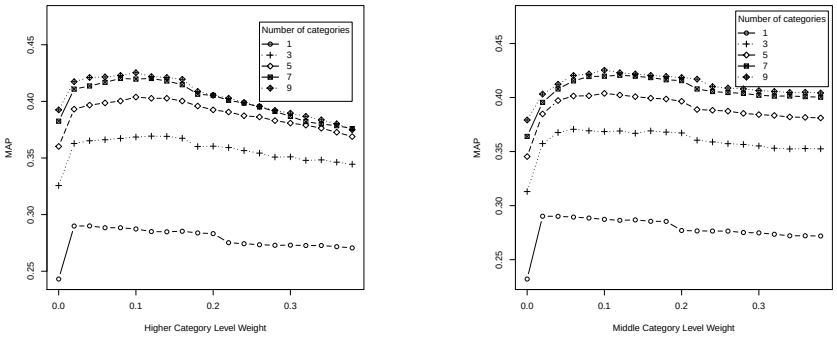
(b) Effect on MAP of using the merchant feature

Figure 6: Retrieval performance (MAP) as influenced by title weight and addition of merchant as a feature

benefits a small number of queries, does not improve the vast majority of queries, and makes worse another set of queries.

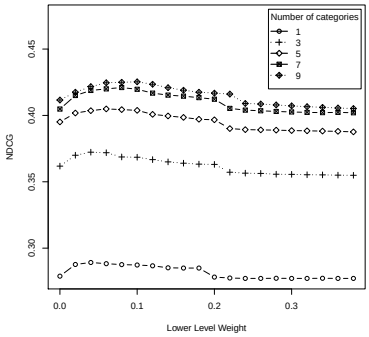
Overall, our best performing model –a combination of all features and heuristics– achieves over a 40% relative improvement over the BM25F baseline in NDCG, and an even larger ($> 200\%$) improvement in MAP. This is significant, given that BM25F gives state-of-the-art results in general text-retrieval. The method is also able to outperform Broder et al.'s [6] method, with differences that are more noticeable at early precision levels. Although we have not collected click-through data in this study, we expect that this level of improvement would measurably impact user activity on the website.

The remaining Figures show in more detail the effect of the parameters. Figure 5 shows the distribution of MAP values across all other parameter settings for 1, 5 and 9 categories. On the x-axis, we show the MAP values, and the y-axis of the plot shows the number of times we get that MAP value as a result (the histogram has been smoothed using a normal kernel). The distribution of their mass (and visually, the height and location of the peaks) tells us that nine categories provide better results on average than using five categories, which is in turn better than using just one category. These results reflect on the probability of obtaining a particular performance value using



(a) First level category

(b) Second level category

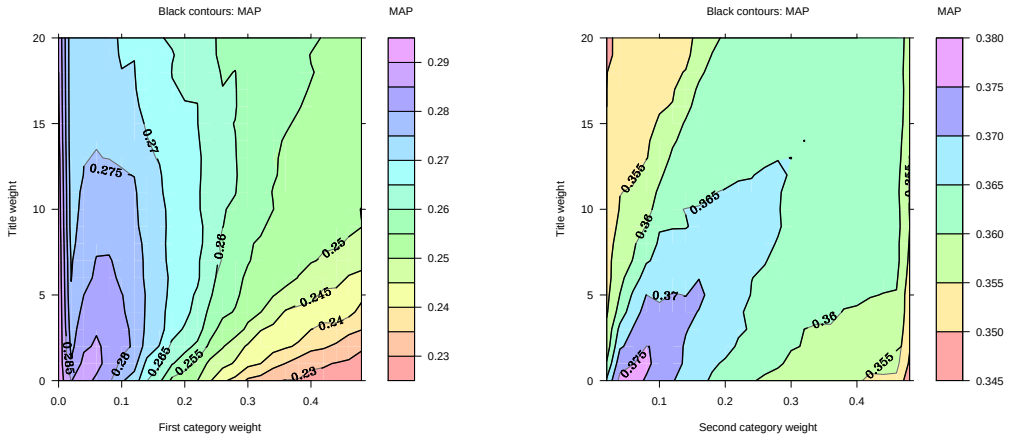


(c) Third level category

Figure 7: Effect on MAP of the category weight for different levels of categorization

the multidimensional search algorithm, which clearly peaks in two different maximum values for all the different number of categories. The plot reveals how assigning a larger set of categories to the deals results in increased retrieval performance, consistently through different parameter configurations. This is a result of being able to weigh properly the different fields, as the gains are consistent for non-graded measures (MAP, P@X).

Figure 6a demonstrates the effect of changing the title weight for different numbers of categories. We can see that the title weight has a small effect above zero and we can again observe that using multiple categories gives better performance than relying on a single category. Figure 6b shows the consistent effect of incorporating the merchant as a feature.



(a) Effect on MAP of varying the title and category 1 weight

(b) Effect on MAP of varying the title and category 2 weight

Figure 8: Retrieval performance (MAP) as a function of category weight and title weight (categories 1 and 2). The x-axis contains different values of the category weights and the y-axis values of the title weights. The heatmap shows the relative influence of each parameter value.

In Figure 7, we show the effect of changing the value of the category weight parameter using different numbers of categories and then observing the average MAP metric. The performance values that we show are averaged over all configurations for all other parameters. The three sub-figures show the effect of the weight at different levels of categorization. Performance is rather robust with respect to changes in the weight, but in all three cases there is a non-trivial i.e., non-zero and non-infinite optimal weight whose value depends on the level of categorization and the number of categories used. In particular, this confirms that it is worthwhile to optimize category weights separately for different levels of classification. In all sub-figures we see a similar difference between the curves i.e., increasing the number of categories leads to improvements up to 9 categories.

Figure 8 shows the inter-dependence of the title and category weights for different levels of categorization. The performance values we show use category weight equals zero for the remaining 2 category levels corresponding to the methods BM25F + Cat1, BM25F + Cat2, BM25F + Cat3 in Tables 2 and 3. We can observe that given one of the parameters, there is a non-zero

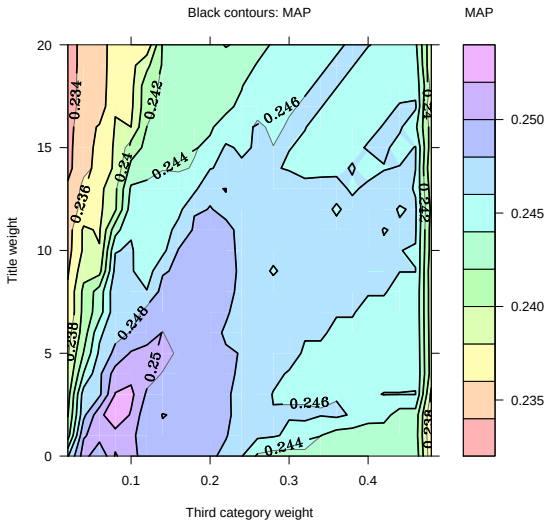


Figure 9: Retrieval performance (MAP) as a function of category weight and title weight (category 3). The x-axis contains different values of the category weights and the y-axis values of the title weights. The heatmap shows the relative influence of each parameter value.

optimal weight for the other parameter in all three plots. This confirms that the model effectively combines relevance signals from keyword and category match. Note that there is a drop in performance (not shown) when the category weight is equal to zero, resulting in a low MAP (the one obtained by BM25F on its own). However, when using the category information the performance ramps up quickly even with low weight values, due to the fact that the extra fields are reducing data sparsity (the method is able to assign a non-zero value to a higher number of documents).

6. Conclusions

We have addressed the problem of ranking daily deals according to their relevance to a user query, a retrieval task that is essential to the commercial success of both deal providers, aggregators and general purpose web search engines that incorporate deals among their results. Compared to the state-of-the-art in text retrieval, we achieve significant improvements on this task using a combination of concept-based and term-based retrieval. In future

work, the model could be enriched by taking other features into account such as social signals (Facebook Likes or comments) or performance data (click popularity or purchase data). Personalization could also improve our results, especially on the deals routing task where long term interests dominate. Nevertheless, we already achieve good results in absolute terms as well.

From a theoretical perspective, we are both intrigued and inspired by the overall success in exploiting concept-based retrieval, even with an automated classifier that was not specifically designed for the task and achieves only moderate accuracy. In a curious way, this takes us back to the roots of information retrieval, where classification (e.g., decimal systems in libraries and later Web directories such as the Yahoo Directory[10]) were the primary way of accessing document collections. We expect that the reasons of success lie partly in the specifics of deals data and the type of queries in the shopping domain as described in Section 3.2. We are thus motivated to find out to what other tasks could be successfully tackled by our method.

[1] G. Amati. *Probability Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, Department of Computing Science University of Glasgow, 2003.

[2] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20:357–389, October 2002.

[3] J. Baron, D. Lewis, and D. Oard. TREC-2006 legal track overview. In *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.

[4] P. N. Bennett, K. Svore, and S. T. Dumais. Classification-enhanced ranking. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 111–120, New York, NY, USA, 2010. ACM.

[5] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. Tran Duc. Repeatable and reliable search system evaluation using crowdsourcing. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 923–932, New York, NY, USA, 2011. ACM.

[10] `dir.yahoo.com`

[6] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 559–566, New York, NY, USA, 2007. ACM.

[7] C. Buckley. Automatic query expansion using smart : Trec 3. In *In Proceedings of The third Text REtrieval Conference (TREC-3)*, pages 69–80, 1994.

[8] J. W. Byers, M. Mitzenmacher, and G. Zervas. Daily deals: Prediction, social diffusion, and reputational ramifications. *CoRR*, abs/1109.1530, 2011.

[9] Y. Chen, G.-R. Xue, and Y. Yu. Advertising keyword suggestion based on concept hierarchy. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 251–260, New York, NY, USA, 2008. ACM.

[10] D. Damljanovic, M. Agatonovic, and H. Cunningham. Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In *The Semantic Web: Research and Applications*, volume 6088, pages 106–120. 2010.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[12] M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic Search Meets the Web. In *IEEE Semantic Computing*, pages 253–260, 2008.

[13] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[14] A. Joshi and R. Motwani. Keyword generation for search engine advertising. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, ICDMW '06, pages 490–496, Washington, USA, 2006. IEEE Computer Society.

[15] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49 – 79, 2004.

[16] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.

[17] Y. Lv and C. Zhai. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 255–264, New York, NY, USA, 2009. ACM.

[18] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[19] D. Osman, J. Yearwood, and P. Vamplew. Automated opinion detection: Implications of the level of agreement between human raters. *Inf. Process. Manage.*, 46(3):331–342, 2010.

[20] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the trec-2011 microblog track. In *In Proceedings of TREC 2011*, 2011.

[21] Y. Qiu and H.-P. Frei. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 160–169, New York, NY, USA, 1993. ACM.

[22] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond, foundations and trends in information retrieval. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[23] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM.

[24] J. J. Rocchio. *Relevance feedback in information retrieval*, volume chapter 14, pages 313–323. Prentice Hall, 1971.

[25] R. S. and W. S. Some simple effective approximations to the 2 poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, New York, NY, USA, 1994. ACM/Springer Verlag.

[26] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 05*, page 43, 2005.

[27] I. Soboroff and D. Harman. Novelty detection: the trec experience. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 105–112, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[28] M. Taylor, H. Zaragoza, N. Craswell, S. Robertson, and C. Burges. Optimisation methods for ranking functions with multiple parameters. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 585–593, New York, NY, USA, 2006. ACM.

[29] X. Wei. *Topic models in information retrieval*. PhD thesis, University of Massachusetts Amherst, 2007. AAI3289216.