Penguins in Sweaters, or Serendipitous Entity Search on User-generated Content

Ilaria Bordino
Yahoo! Research
Barcelona, Catalunya, Spain
bordino@yahoo-inc.com

Yelena Mejova
Yahoo! Research
Barcelona, Catalunya, Spain
ymejova@yahoo-inc.com

Mounia Lalmas
Yahoo! Research
Barcelona, Catalunya, Spain
mounia@yahoo-inc.com

ABSTRACT

In many cases, when browsing the Web users are searching for specific information or answers to concrete questions. Sometimes, though, users find unexpected, yet interesting and useful results, and are encouraged to explore further. What makes a result serendipitous? We propose to answer this question by exploring the potential of entities extracted from two sources of user-generated content – Wikipedia, a user-curated online encyclopedia, and Yahoo! Answers, a more unconstrained question/answering forum – in promoting serendipitous search. In this work, the content of each data source is represented as an entity network, which is further enriched with metadata about sentiment, writing quality, and topical category. We devise an algorithm based on lazy random walk with restart to retrieve entity recommendations from the networks. We show that our method provides novel results from both datasets, compared to standard web search engines. However, unlike previous research, we find that choosing highly emotional entities does not increase user interest for many categories of entities, suggesting a more complex relationship between topic matter and the desirable metadata attributes in serendipitous search.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous

Keywords

Entity Search, Entity Networks, Serendipity, Interestingness, Metadata

1. INTRODUCTION

Why do penguins wear sweaters? An unsuspecting user may stumble on the answer to this question while researching *oil spills*, finding a song about penguin sweaters made to help rehabilitate penguins injured by oil spills in Tasmania. Such surprises are welcome in *serendipitous search*, which occurs when a user with no a priori or totally unrelated intentions interacts with a system and acquires interesting information [39]. A system supporting serendipity must provide results that are *surprising*, *semantically cohesive*, i.e., *relevant* to some information need of the user, or just *interesting*. In this paper, we tackle the question of what makes a result serendipitous.

To this end, we examine two of the largest user-generated knowledge repositories: Yahoo! Answers and Wikipedia. Yahoo! Answers is nowadays one of the largest community question/answering systems, with millions of users posting millions of questions and hundreds of millions of answers.[1] A study reported in [27] suggested that while Yahoo! Answers is not optimal for factoid search, it is becoming the destination of choice for complex information needs such as opinion or advice, making it the perfect source to investigate serendipitous search. Wikipedia, on the other hand, is a popular collaboratively-edited online encyclopedia that employs a staff of editors and an army of volunteers to maintain the quality of its articles. The highly curated nature of Wikipedia may make it a more trustworthy source of information. However, the freedom of conversation on Yahoo! Answers presents its own advantages, containing within it opinions, rumors, and social interest and approval.

Some previous attempts have been made to introduce serendipity into browsing systems having a social aspect, such as TweetMotif [33] for exploring Twitter and Auralist [42] for recommending music. However, none of these have rigorously defined, operationalized, and evaluated user-generated content-driven serendipitous search. In this paper we develop an entity-based exploratory search framework that represents the content of each data source as an entity network. We describe some of the challenges of extracting entities from these two different sources, as well as building a meaningful similarity measure for entities. Our entity-retrieval algorithm, based on lazy random walk with restart, achieves 67% accuracy on Wikipedia and 72% on Yahoo! Answers (as assessed using crowd-sourcing), putting it on par with similar recommendation systems [6, 7, 8].

Following [16], we delve further into what makes search serendipitous by using metadata of the documents in both collections to compute summary statistics for each entity. This way, we estimate (i) the intensity of the emotion, (ii) the quality of the writing, and (iii) the topical category of the text surrounding each entity. By constraining the result sets using these statistics we measure the extent to which each dimension contributes to the perceived serendipity.

CIKM '13 San Francisco, CA, USA
ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505680 .

[1] http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served/

We take two approaches to assessing serendipity of search results. The former, proposed by Ge et al. [15] in the context of recommender systems, considers two main attributes of serendipity: unexpectedness (or surprise) and usefulness. The unexpectedness factor is computed by comparing to some "obvious" baseline, while the usefulness can be estimated using standard relevance judgments.

In the second approach we go beyond relevance by also considering the interestingness of the results. Previously, Andre et al. [2] evaluated web search results in terms of their relevance and "interestingness", hypothesizing that "search results that are interesting but not highly relevant indicate a potential for serendipity." Using crowd-sourcing we conduct a set of experiments that concern various kinds of entities, such as people, places, events, websites, gadgets, sports, and health-related topics. Our experiments include tens of thousands evaluations.

The remainder of this paper is organized as follows. Section 2 describes related work and positions our approach. In Section 3 we introduce the datasets and the methodology applied to construct the entity networks. Section 4 presents the retrieval algorithm, while Section 5 describes and analyzes the metadata extracted to supplement the entity networks. In Section 6 we report the performance of our retrieval method. Finally, Section 7 presents our two approaches for evaluating the serendipity of the retrieved results, analyzing the effect of metadata constraints. We end the paper with conclusions and thoughts for future work.

2. RELATED WORK AND BACKGROUND

General-purpose search engines have been extended in various ways to support exploration and promote serendipity, for instance, using corpus-wide analysis [33], adding a temporal dimension [37], and by incorporating domain-specific resources [9]. Some attempts have been made to characterize the interestingness of documents. For instance, when browsing news stories, O'Brien [32] finds that people clicked on articles that were weird, odd, and shocking, even though they were not necessarily interested in reading about the topic. We go one step further, by exploiting what internet users may consider interesting *because* they are *writing* about it. We do so within the context of *entity search*.

Many useful facts about *entities* (people, locations, organizations, or products) and their relationships can be found in data sources such as Wikipedia or Freebase. Others need to be extracted from unstructured text, as is the case with Yahoo! Answers. The problem of discovering interesting relations from unstructured text has led to a surge in research on *entity search* [4, 11, 17, 26, 29, 30, 34], along with evaluation efforts like INEX Entity and Linked Data tracks,[2] TREC Entity track,[3] and SemSearch challenge.[4] An entity search system requires to extract entities, measure the proximity between two entities, and rank entities according to their proximity to a query entity.

For entity extraction, we follow the common approach that for an extracted entity to exist, it must appear as a Wikipedia page [17, 26, 29, 30, 34]. The problems of measuring entity similarity, and retrieving entities related to an input entity, have been tackled in several works [10, 11, 19, 24] by building graphs of entities and their relations, and applying random-walk computations [12, 20] on these graphs. We adopt a similar approach. We extract entities from Yahoo! Answers and Wikipedia, and build entity networks based on the textual similarity of the documents where entities appear. We also enrich our networks with various metadata.

Other approaches [1, 41] build *entity-relationship* models, where entities take part (with various roles) in different types of relations representing real-world associations. These semantically richer models require the usage of structured query languages. Although interfaces supporting such queries exist ([41]), we target non-expert, every-day users of social media. Moreover, we do not at this stage wish to rely on any visualization paradigm. A graph of pairwise relations is a natural choice to model entity similarity in our context.

In terms of algorithms, as our focus is on what makes results relevant and interesting, we use random-walk methods (state of the art for recommendation problems [7, 8, 20]).

A recent workshop Searching4Fun[5] focusing on "pleasure-driven, rather than task-driven, search", has called for more studies looking at "what makes users happy" in this type of search [23]. One proposal put forward in [16] is to extract documents that (*i*) contain unexpected nuggets of information, (*ii*) evoke emotional meaning using sentiment analysis, and (*iii*) contain useful knowledge as identified by user-generated metadata. In this paper, we explore similar dimensions of our datasets to understand what makes a data source interesting and which *associated metadata* (sentiment, quality and topicality) promotes serendipity and to what extent.

In this paper, we use implicit metadata – those extracted from the documents. We focus on quality of the writing, sentiment strength, and topical category – all extracted from the text in the collections. Text categorization is a well known IR task, and many algorithms have been developed to automatically classify documents according to some given taxonomy [36]. Non-topical implicit metadata have also been experimented with, for example related to the quality, credibility, and emotion of the text. Because of its applicability across data sources and its previous use in the context of web search [22], we use readability as our quality metadata. Also, sentiment analysis has become essential for social media-driven applications [21], whether for monitoring purpose or as additional feature. Its role in generating interesting results remains to be examined.

Our aim is to provide insights to what makes a result serendipitous, in the context of entity search. Although the visualization paradigm, i.e. the display of the search results, will affect how users experience serendipity, this work focuses on the search results. We leave for future work aspects concerned with evaluating the user experience.

3. ENTITY NETWORK EXTRACTION

3.1 Datasets

Yahoo! Answers. The largest community-driven question/answering web portal, Yahoo! Answers was launched in 2005. The portal allows people to ask questions on different topics and answer questions asked by other users, sharing their knowledge and opinions. Every question is assigned

[2] http://www.inex.otago.ac.nz/tracks/entity-ranking/entity-ranking.asp
[3] http://ilps.science.uva.nl/trec-entity/
[4] http://semsearch.yahoo.com/
[5] fitlab.eu/searching4fun/schedule.php

by the asker to one category in a hierarchy of categories. This manual classification of questions into topics is meant to help answerers, who typically find questions by browsing or searching the category hierarchy. We collected a set of Yahoo! Answers documents from 2010-2011. We extracted the English-language questions, and the answers to these questions. Our dataset, dubbed YA, consists of 67 336 144 questions and 261 770 047 answers.

Wikipedia. Wikipedia[6] is a multilingual, web-based, free-content encyclopedia, written collaboratively by a large number of volunteers. Since its creation in 2001, Wikipedia has grown into one of the largest reference websites, attracting 470 million visitors monthly as of February 2012. As of September 2012, there are more than 77 000 active contributors working on over 22 000 000 articles in 285 languages. We use the English Wikipedia dump[7] from December 1, 2011, which consists of 3 795 865 articles. We use WikiExtractor[8] to strip the meta-content and extract the text. In the remainder of the paper, we dub this dataset WP.

Data availability. Our two datasets consist of public data. Dumps of Wikipedia are publicly available,[9] and Yahoo! Answers data can be collected using a crawler.

3.2 Entity Extraction

We call *entity* any concept that is well defined and described in a Wikipedia page. Given a piece of text, we first parse the text to identify surface forms that are candidate mentions of Wikipedia entities. We add entity candidates to each recognized phrase by retrieving the candidates from an offline Wikipedia database.

To resolve each surface form to the correct Wikipedia entity we apply the machine-learning approach proposed by Zhou et al. [43]. This approach employs a resolution model based on a rich set of both context-sensitive and context-independent features, derived from Wikipedia and various other data sources including web-behavioral data. The authors report that the model achieved 85% precision and 87.8% recall when evaluated on a manually-labeled set of news articles. We then use Paranjpe's *aboutness* ranking model [34] to rank the obtained Wikipedia entities according to their relevance for the text. This model exploits structural and visual properties of web documents, and user feedback derived from search engine click logs. The method achieved 75% accuracy when evaluated against a ground truth of editorial relevance judgements for a collection of query-url pairs. Paranjpe has shown that his approach, even when trained mainly on head web pages, generalizes and performs well on all kinds of documents, including tail pages.

We are aware of the existence of more recent entity extraction tools, such as Wikipedia Miner[10] and Tag Me[11], which have been shown to outperform previous approaches. However, some technical issues that we had to face while dealing with our large-scale datasets, made us favor the method describe above. We remark that detecting and disambiguating entities that are mentioned in documents is not the objective of this work. We believe that improving the entity-extraction step will probably lead to improving the overall performance of our system. We leave this for future work.

For consistency, we apply the extraction methodology above to both Wikipedia and Yahoo! Answers; in the YA dataset we apply the algorithm on both questions and answers. In the WP dataset, entities could also be extracted using inter-wiki links associated with the surface forms in the articles. However, such linking is not consistent, and it is often done for only the first few appearances of the entity in an article.

3.3 Entity Similarity

Using the methodology described above, we extract 896 799 distinct entities from YA, and 1 754 069 from WP.

Similarity Measure for Entities. Starting from the set of entities extracted from each dataset, we construct an entity network by using a content-based similarity measure to create arcs between entities. We first build a textual representation e_C of any entity e extracted from a document collection C, by taking the (order-insensitive) concatenation of all the documents in C where entity e appears. We dub *entity document* such textual representation of an entity.

Let E_C be the set of entity documents of the entities extracted from a collection C. Moreover, let L_C be the lexicon of C. We extract the lexicon by tokenizing every document, removing stop words and applying Porter's stemming algorithm on the obtained tokens [28].

We apply on the set E_C of entity documents extracted from a collection C the vector-space model [35]. More precisely, we extract from each entity document $e_C \in E_C$ a $|L_C|$-dimensional vector v_{e_C}, where each dimension represents a term in the lexicon of the collection. Using the well-known TF/IDF scheme, we assign the following weight to term dimension i in the vector representation of e_C:

$$v_{e_C}[i] = tf_{i,e_C} \cdot \log \frac{|E_C|}{|\{e_C \in E_C : i \in e_C\}|}$$

where tf_{i,e_C} and $idf_{i,e_C} = \log|E_C| - \log|\{e_C \in E_C : i \in e_C\}|$ respectively represent the frequency of term i in the entity document e_C, and the inverse document frequency of i in the collection E_C of entity documents.

Once we have created a TF/IDF vector representation of all the entities in a dataset C, we adopt the cosine distance to measure the similarity between two entities. Our arc-weighting function is thus the following:

$$w_C(e,f) = cos(v_{e_C}, v_{f_C}) = \frac{\sum_{0 \le t \le |L_C|} v_{e_C}[t] \cdot v_{f_C}[t]}{||v_{e_C}|| \cdot ||v_{f_C}||}$$

Because the TF/IDF weights cannot be negative, the similarity values will range from 0 to 1. Given that cosine distance is a commutative function, we create an undirected network by computing all the pairwise similarities between the entities in a collection. However we do not build a complete graph. Instead we connect with an arc only the pairs that achieve a similarity value higher than a minimum threshold σ. This standard pruning strategy is used to avoid considering poorly significant relations [5].

Implementation. Building the entity-network representation of each dataset requires the computation of all pairwise cosine similarities among the entities in the dataset. Performing an all-pairs similarity computation is a challenging task when one has to deal with datasets of very large scale, because the number of potential candidates to evaluate is

[6] wikipedia.org
[7] http://dumps.wikimedia.org/enwiki/20111201/
[8] medialab.di.unipi.it/wiki/Wikipedia_Extractor
[9] http://dumps.wikimedia.org/
[10] http://wikipedia-miner.cms.waikato.ac.nz
[11] http://tagme.di.unipi.it

quadratic in the number of nodes and thus can be enormous. We have extracted millions of entities from YA and WP, so our problem instances are exactly of this type.

We first reduce the candidate space by restricting to the pairs of entities that co-occur in at least one document. To solve the problem efficiently, we then perform the all-pairs similarity computation by applying the algorithm of Baraglia et al. [5]. The algorithm is a distributed algorithm that works in the Hadoop[12] framework, so as to exploit the aggregated computing and storage capabilities of large clusters. Scalability is achieved by embedding state-of-the-art pruning techniques, as well as introducing a partitioning strategy able to overcome memory bottlenecks.

3.4 Entity Networks

We extract an entity network from each of our two datasets, using the arc-weighting function described above, and setting the minimum similarity threshold to $\sigma = 0.5$. This value was chosen heuristically in a preliminary assessment of the quality of the similarity measure built. Table 1 provides a basic characterization of the two networks.

Node overlap. The YA network contains 51% of the nodes in the WP network. The fact that the number of entities extracted from Yahoo! Answers is smaller than the one obtained from Wikipedia is clearly related to the different nature (non-curated vs. curated) of the two datasets. In Yahoo! Answers, many questions and answers are extremely short, and contain some quick exchange of communication where no entities occur. Instead, Wikipedia documents provide a wealth of useful mentions of other entities that are relevant for the entity that is the subject of the page.

Connectivity. The two graphs are almost fully connected. The largest connected component (CC) spans 92.15% of the nodes in YA, and 95.78% in WP. This is due to the presence of popular entities that appear ubiquitously in the two datasets. These entities represent very common concepts, which are not particular to the subject of a document.

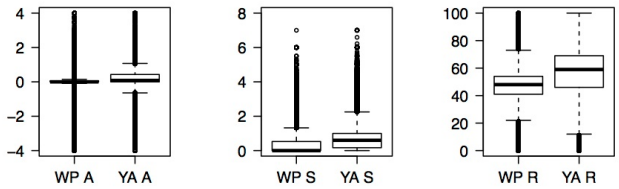
Availability. To facilitate reproducibility of our experiments, we make our entity networks available upon request.[13]

4. RETRIEVAL

Algorithm. Random-walk based algorithms such as Personalized PageRank [20] or center-piece subgraphs [40] have been applied in many recommendation problems [6, 7, 8, 12]. Our algorithm for extracting from a network the top n entities that are most related to a query entity, is inspired by the above research line. More specifically, the algorithm performs a *lazy* random walk with restart to the input entity.

Our method takes as input a graph, a self-loop probability β, and a start vector defined on the nodes of the graph, which in this case contains only the input entity. The random walk starts in the node corresponding to such entity. At each step, it either remains in the same node with probability β, or follows one of the out-links with probability $1 - \beta$. In the latter case, the links are followed with probability proportional to the weights of the arcs. Self-transitions are inserted to reinforce the importance of the starting node, by slowing diffusion to other nodes. The value of the self-loop probability is set to $\beta = 0.9$, following previous works on query recommendation [6, 12]. We do not use random jumps, because by setting the random-jump probability to the standard value of $\alpha = 0.15$, we noticed a worsening of the results. As stopping criterion, we check whether the norm of the difference between two successive iterations is $< 10^{-6}$, or we stop the random walk after a maximum of 30 iterations. We implement the algorithm by customizing the PageRank implementation contained in the LAW[14] library.

Figure 1: Distribution of attitude (A), sentimentality (S), and readability (R) for YA and WP



Scoring method. Our scoring method basically ranks the entities based on the stationary distribution of the lazy random walk described above. However, we noticed that popular entities with very large degree appear ubiquitously in the prominent positions of the ranking vectors of all the entities in our datasets. We introduce two corrections for this.

First, we measure the rarity of any entity e in a data collection C by computing its inverse document frequency $IDF(e) = \log(N) - \log(DF(e))$, where N is the size of collection C, and $DF(e)$ is the document frequency of entity e. Given the ranking vector obtained for an input entity, we filter out the top M entities with lowest inverse document frequency. These entities represent common concepts that appear in the majority of the documents, and thus are not likely to be relevant to the input entity. The value of M is heuristically set to 500 in YA, and to 1 000 in WP.

For our second correction method, we divide the ranking vector by the global PageRank values obtained by using no personalization (that is, starting at random at any node), and fixing the random jump probability to $\alpha = 0.15$. In our experiments we obtain the best results with normalization by the squared root of global PageRank scores.

5. METADATA

We extract from our dataset information regarding quality, sentiment, and topical categories. The selection of the metadata was influenced by the "Searching4Fun workshop (see Section 2), the fact that their extraction could be automated using known tools, and our intuitions about their potential to operationalize serendipity. The metadata features are first collected at document/sentence level, and then aggregated to derive scores for all the entities in a dataset.

Quality. We can derive quality measures for our document collections in several ways. In Yahoo! Answers we could leverage explicit user feedback, such as stars on questions, or best-answer ratings for answers. In the case of Wikipedia, we could count *dispute* alert messages used by Wikipedia editors as an indication of poor quality of a document. An alternative that is widely applied in web search is *readability*, which provides an indication of the difficulty that

[12] `hadoop.apache.org`

[13] Please email `bordino@yahoo-inc.com`

[14] `law.di.unimi.it/software.php`

Table 1: Basic characterization of the networks extracted from Yahoo! Answers (YA) and Wikipedia (WP)

Dataset	# Nodes	# Edges	Density	# Isolated	Avg Degree	Max Degree	Size of Largest CC
YA	896 799	112 595 138	0.00028	69 856	251.10	231 921	826 402 (92.15%)
WP	1 754 069	237 058 218	0.00015	82 381	270.30	346 070	1 671 241 (95.28%)

a reader may encounter in comprehending a text. Lower readability scores are assigned to more sophisticated documents, which require higher education level to be understood. We choose readability, because of its applicability on both datasets. For each document we compute the Flesch Reading Ease score [14]. We then derive a readability score for every entity by computing the median Reading Ease over all the documents where the entity appears. Figure 1 shows the distribution of the readability scores for entities in the two datasets: observe that Yahoo! Answers entities tend to have higher readability scores, indicating that they were extracted from documents that were easier to understand.

Sentiment. We classify the documents in both datasets using SentiStrength,[15] a state-of-the-art tool for extracting positive and negative sentiment from informal English text. SentiStrength has been shown to outperform several (un)supervised alternative approaches on a number of different social web data sets, including MySpace, Twitter, YouTube, Digg, RunnersWorld, BBCForums [38]. By default, the tool computes document-level sentiment scores: each sentence within a document receives two scores from 1 to 5 — for positivity and negativity — and then the scores are averaged over the document. However, a document is likely to mention many different entities, and the sentiment expressed around them may vary considerably from entity to entity. To obtain *entity-level* scores, we first compute sentiment for each mention of an entity in a document, by considering a small window of text around the mention (we include the ten words preceding the mention, and the ten words following it). We further calculate *attitude* and *sentimentality* metrics [25], which measure "the inclination towards positive or negative sentiments" and the "amount of sentiment," respectively. Finally, for every entity we compute average attitude and sentimentality over all the text segments – extracted from different documents – where the entity is mentioned. Figure 1 shows the distribution of entity-level attitude and sentimentality in the two datasets: notice that Yahoo! Answers entities tend to have higher attitude and higher sentimentality, reflecting how the Q&A forum contains a broader expression of opinions and emotions with respect to Wikipedia, which tends to be more neutral.

Topic Categories. Both datasets have their own categorization system. In Yahoo! Answers each question is assigned exactly one category chosen by the asker. Every answer to a question is listed under the same category of the question. Wikipedia pages are also organized in a hierarchical category structure. However, for consistency and comparability of the results obtained by the two different media, we decide not to use these dataset-specific categories, and to refer to an external categorization. Specifically, we use a proprietary system developed to support automated categorization of various data sources, such as news articles, tweets, web pages and RSS feeds. Our classifier relies on a proprietary taxonomy, designed by an editorial team responsible for maintaining category definitions clear, distinct, organized, and stable in the face of constantly changing data-source and performance requirements. The taxonomy consists of various sub-taxonomies that cover particular categorical facets, such as *People, Organizations, Regions, Events*, and a number of main *Subjects* listed in Table 2.

Our classifier has been trained on a corpus of US-English news articles and tested on various kinds of datasets, achieving a micro-precision at 80% coverage of 92.5% on news data, 82% on RSS feeds, and 70% on Wikipedia data. The classifier annotates each document with three topical categories. To derive entity-level topical features, we assign to an entity the three most frequent categories associated with the documents where the entity occurs.

Notice that our choice of adopting a proprietary classifier was driven by the practical need for a system that could be deployed on Hadoop so as to handle the automatic classification of datasets of very large scale. In future work we plan to extend our method to public taxonomies, such as ODP (`www.dmoz.org/`) and Yahoo! Directory (`dir.yahoo.com`).

Table 2: Main subjects considered for categorization

Arts & Entertainment	Beauty
Business	Education
Family & Relationships	Finance
Food & Cooking	Health
Hobbies & Personal Activities	Home & Garden
Nature & Environment	Politics & Government
Real Estate	Science
Society & Culture	Technology & Electronics
Transportation	Travel & Tourism

6. RETRIEVAL PERFORMANCE

Testbed. We tested the performance of our system using a set of test queries. First, we collected the most searched queries in 2010 and 2011 from Google Zeitgeist.[16] Subsequently we found Wikipedia pages for each of those queries, that is, we identified the entity associated with each query. We finally checked the coverage of these entities in both datasets, YA and WP. Coverage here is defined as the number of documents mentioning each entity. We included in our test set the top 50 queries with highest coverage. The resulting 50 queries encompass a diverse set of topics, such as people, places, websites, events, gadgets, sports, and health.

Performance. For each query, we retrieved related entities from the YA and WP entity networks. We then used `CrowdFlower.com` – a virtual marketplace for micro-tasks – to assess the relevance of the top 5 results retrieved for the queries in our testbed. To ensure quality, we built a set of *gold standard* query-result pairs. Contributors were required to complete a preliminary training session, in which they were shown six golden questions and had to provide correct

[15] `sentistrength.wlv.ac.uk`

[16] `www.google.com/zeitgeist`

answers for at least four of them, before they were granted access to our task. Moreover, golden questions were also randomly inserted in the real task, and used as a hidden test to check the quality of a contributor's performance. Whenever a contributor's accuracy on the gold standard dropped below 70%, the contributor was considered untrustworthy and his or her judgements were discarded. In total, 1 587 query-result pairs were labeled, with 3 annotations per task.

Because a large number of labelers were working on largely disjoint sets of tasks, instead of a standard Cohen's kappa we report label overlap between the participants, which is 85% overall. The lowest agreement was on advanced topics involving generally unfamiliar entities, such as *Secosteroid* (a kind of molecule), and those involving thorough understanding of an issue, such as the politician *Sally Kern* retrieved for the query entity *Terrorism*.

Table 3 shows the number of relevant entities out of top 5 returned. On average, our algorithm performs with a precision of 66.8% on WP and 72.4% on YA. These accuracy values are comparable with those achieved by recent works on recommendation problems [6, 7, 8].

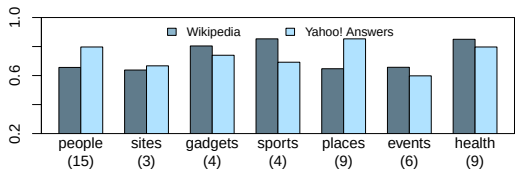
The algorithm performs somewhat similarly on the two datasets, with 0.41 correlation between their performances. Perfect results are achieved for 18 (WP) and 13 (YA) result sets, with 16 (WP) and 9 (YA) result sets having fewer than 3 relevant entities returned. When we examine the ranking performance of our algorithms by comparing the Mean Average Precision scores – 0.716 (WP) and 0.762 (YA) – to precision, we see an improvement in scores, indicating that the relevant entities tend to be shown at the top of the rankings. Figure 2 shows the MAP scores of the runs grouped by category. Wikipedia tends to perform well on topics such as major events and sports, whereas Yahoo! Answers does better with people and places.

Table 3: # relevant entities retrieved in the top 5

	WP	YA		WP	YA
Justin Bieber	1	4	Tennis	2	3
Nicki Minaj	0	4	Mount Everest	3	5
Katy Perry	4	5	Eiffel Tower	0	4
Shakira	5	4	Oxford Street	2	3
Eminem	5	5	Nérburgring	3	3
Lady Gaga	5	4	Haiti	5	4
José Mourinho	4	4	Chile	5	5
Selena Gomez	2	5	Libya	5	4
Kim Kardashian	3	4	Egypt	5	4
Miley Cyrus	3	5	Middle East	3	5
Robert Pattinson	1	2	Earthquake	4	5
Adele (singer)	0	0	Oil spill	2	2
Steve Jobs	5	4	Tsunami	5	3
Osama bin Laden	5	5	Subprime mortgage crisis	4	3
Ron Paul	2	5	Bailout	0	1
Twitter	4	5	Terrorism	4	2
Facebook	2	4	Asperger syndrome	4	3
Netflix	1	1	McDonald's	1	4
IPad	4	5	Vitamin D	4	3
IPhone	5	4	Appendicitis	3	2
Touchpad	5	4	Cholera	5	3
Amazon Kindle	1	1	Influenza	5	5
Olympic Games	5	2	Pertussis	2	4
Cricket	5	4	Vaccine	4	4
FIFA	5	4	Childbirth	5	3

Rank aggregation. Although the two datasets have comparable performance, the overlap between the results is very small – an average of 0.6 entities (that is under one result) in common in the top 5. Gadgets, places, and websites have the most overlap of 1.33, 0.78, and 0.75 result per query, respectively. These include popular topics like *Facebook*. However, most of the results do not overlap, suggesting that combining the results would improve recall, and perhaps introduce more diversity.

Figure 2: Mean Average Precision of entity retrieval by category (# query entities in parentheses)



To verify this hypothesis, we build a global ranking of the results extracted from the two datasets by applying the simple median-rank aggregation schema proposed by Fagin et al. [13]. We consider each of the two top-n ($n = 5$) result lists as a *partial ranking* of the recommendations appearing in the union of the two lists, where all the items not appearing in a list form a tie with rank $n + 1$. To aggregate the two partial rankings, we sort the full set of recommendations produced for an entity according to the median of the ranking scores obtained in the single lists (in this simple case there are only two input rankings, so the median score coincides with the mean). This algorithm provides a constant-factor approximation solution [13] for the problem of aggregating partial rankings with ties. The aggregation of the two rankings achieves an accuracy of 74.4% and a Mean Average Precision of 0.782, improving the performance on the individual datasets.

Error analysis. A further testament to the efficacy of our retrieval algorithm is the fact that, upon manual inspection, we do not find relevant entities in the immediate neighborhood of the query entity. For example, *Egypt*, an entity for which both runs produce good results, has *British Pacific Fleet* and *FC Groningen* (a football club in the Netherlands) as the top two closest entities in the WP network, and *Spring (device)* and *IGN* (an entertainment website) in the YA network. The case of the entity *Spring (devise)* is especially indicative, since it may have been mistakenly detected in text mentioning the "Arab Spring" events, but our algorithm has then downgraded it for the lack of relation with other entities dealing with Egypt.

However, some queries have resulted in non-satisfactory performance. In a few cases, our method has retrieved a list of entities that are highly similar to each other (albeit all relevant to the query); for example, different subtypes of the Influenza virus, or versions of the Apple PowerBook. We believe that a result list with little or no diversity is not of great interest for the user. Such behavior probably happens when our random-walk based algorithm becomes trapped in some small and dense component of the entity network.

Second, an entity may be linked to a tightly knit, but not relevant, neighborhood due to errors of the entity extractor, or to noise in the similarity measure used to build the network, which is based on textual similarity of the content from which the entities were extracted. For example, for the query entity *Adele (singer)*, a connection has been made to a totally unrelated result, a famous portrait of Adele Bloch-Bauer, hurting performance. Homonymy can cause errors when the similarity between entities is only syntactic, and

not semantic. These challenges should be addressed both by improving entity extraction and better informing the retrieval algorithm; we leave these for future work.

7. SERENDIPITY & INTERESTINGNESS

We conduct an extensive study to compare the results extracted by our retrieval algorithm from the two datasets (YA and WP), with the goal of understanding what these two different sources of user-generated content can provide to serendipitous search. We consider a basic scenario in which, for our 50 queries, we compare the results extracted from the two entity networks (YA and WP). Second, we attempt to verify which features make the search results more valuable. To this aim we exploit the metadata extracted from the two datasets to enrich the entity networks, and we constrain the retrieval in the dimensions of sentimentality,[17] quality and topical category. For each of the two datasets we build five additional experimental setups in which the results extracted from the general, unconstrained network, are compared to those obtained after introducing a constraint on a specific metadata dimension. To attain the latter, we filter the results of the original retrieval so as to select the top results that satisfy the constraint. The constraints are:

- **Topic** *Question:* Do entities that are topically coherent with respect to the query provide better results? *Constraint 1:* Restrict to the entities that share at least one topical category with the input query.
- **High/ Low Sentimentality** *Question:* Do entities which convey more (less) emotion provide better results? *Constraint 2 (3):* Restrict to the entities with sentimentality score higher (lower) than the median (0.6 for YA, 0 for WP).
- **High/Low Readability** *Question:* Do entities with higher (lower) readability scores provide better results? *Constraint 4 (5):* Restrict to the entities with readability score higher (lower) than the median value (46 for YA, 41 for WP).

In the remainder of this section we first examine how the constrained setups perform with respect to relevance of retrieved results (Subsection 7.1). Next we compare these alternative runs from various points of view. First we consider a notion of *serendipity* as measured in terms of the fraction of *unexpected* results provided by a recommender algorithm, which are also *relevant*. In Subsection 7.2 we compare our experimental setups with respect to this metric. Next we attempt to evaluate other, more subjective aspects of serendipitous search, such as personal *interestingness* to the user, or interestingness with respect to the input query. This analysis is described in Subsection 7.3.

7.1 Constrained-Retrieval Performance

We evaluate the relevance of the results after they have been passed through the metadata filters. Table 4 shows the precision at 5 for each run and marks the significance of the difference from the unconstrained run. Besides YA and WP, we report a third case, dubbed COM, which represents for each (unconstrained or constrained) scenario, the aggregation of the results obtained from YA and WP through the median-rank aggregation scheme described in Section 6. Again, we see the combined results outperform the ones from individual datasets. Low-sentimentality and low-readability constraints negatively affect the performance, however we show later (Section 7.3) that they still can improve result interestingness for certain topical categories.

[17]We focus on sentimentality because attitude provided a much weaker signal in preliminary experiments.

Table 4: P@5 for constrained retrieval (significance of the difference from unconstrained run: * $p < 0.05$)

	WP	YA	COM
Unconstrained	0.668	0.724	0.744
Topic	0.676	0.732	0.760
Low sentimentality	0.460*	0.536*	0.504*
High sentimentality	0.656	0.700	0.736
Low readability	0.488*	0.560*	0.556*
High readability	0.588*	0.732	0.708*

7.2 Serendipity

Recent works [15, 33, 42] have shown that, beyond accuracy, there are many other metrics that can be used to assess the performance of recommender algorithms. Serendipity takes into account the novelty of recommendations and how far recommendations may positively surprise users. To compare our exploratory-search setups in terms of serendipity, we adopt a metric introduced by Ge et al. [15], designed to capture two essential aspects of serendipity, *unexpectedness* and *relevance*.

Given the set RS of recommendations generated by a recommender system, Ge. et al define the set $UNEXP$ of *unexpected* recommendations as the recommendations in RS that are not generated by a baseline prediction model PM. Let *rel* be a function used to assess the relevance of a recommendation. Ge et al. calculate the *serendipity* of set RS as the fraction of unexpected results, which are also relevant:

$$SRDP(RS) = \frac{\sum_{i \in UNEXP} rel(i)}{|UNEXP|}$$

For relevance, we use the editorial judgements collected through the annotation task described in Section 6. Unexpectedness is measured by comparison with benchmarks that produce expected recommendations. We use four baseline generators of obvious recommendations:

1. **Top**: For each query, we retrieve the 5 entities that occur most frequently in the top 5 search results provided by two major commercial search engines;
2. **Top Nwp**: Similar to previous case, but excluding the Wikipedia page of the input entity (if present) from the set of results returned by the search engines. Here the idea is to verify if the Top baseline induces any kind of bias towards the WP dataset;
3. **Rel**: Return the top 5 entities in the related-query suggestions provided by two major search engines;
4. **Top + Rel** Return the union of the sets of entity recommendations provided by Top and Rel.

We discard all the baseline entities that are too recent with respect to the time frame spanned by our entity networks. Keeping those recent entities might induce an unwanted bias in the results, as it would determine higher unexpectedness.

Table 5 reports the value of the serendipity metric computed for each setup, with respect to each baseline. We

report results for YA, WP, and for their aggregation COM. Observe that all of our experimental setups achieve higher serendipity when compared to *Rel* baseline, as opposed to when they are compared to *Top*. The topic-constrained setup outperforms the other setups in almost every baseline/dataset combination. Results comparable with the topic-constrained run are achieved in the unconstrained case, and also in the high-sentimentality and high-readability run. The low-sentimentality and low-readability setups perform considerably worse, due to the fact that these constraints seriously hurt relevance, as reported in Table 4.

We also remark that YA always outperforms WP, achieving a value of serendipity which is typically $6\% - 7\%$ higher. The difference is higher ($10\% - 15\%$) in the readability runs. The best results, with respect to all baselines, are achieved by the combination (COM) of YA and WP.

It is also interesting to notice that results do not degrade – in fact they slightly improve– when we discard (*Top Nwp*) from the set of documents used to build the *Top* baseline, the Wikipedia page corresponding to the input query.

The Rel+Top baseline, which builds a larger pool of entity recommendations, is obviously the strongest baseline, compared to which every setup achieves the smallest fraction of serendipitous results.

Finally, the values in parentheses in Table 5 indicate the fraction of unexpected and relevant results computed with respect to the total number of recommendations extracted by each setup (and not with respect to the sole unexpected recommendations, as in the serendipity metric). Observe that this fraction is always almost as high as the corresponding serendipity value. This confirms that we are indeed retrieving a considerable fraction of results that are both unexpected and relevant, even when compared to the most robust *Rel + Top* baseline.

7.3 User-Perceived Quality

We next attempt to evaluate other more subjective aspects of serendipitous search by performing another set of crowd-sourced evaluations. Besides being relevant to the query, the results must be interesting enough to the user to catch his or her attention, and to encourage further exploration. Although highly subjective, the dimension of *interestingness* has been used to measure the serendipity of web search results [2] and recommender systems [18]. To make sure we separate intrinsic interestingness of entities from the extent to which a user interested in a search query is interested in a presented result, we ask labelers to consider both questions. Furthermore, we attempt to measure the value of the results by asking whether the result allowed one to learn something new about the query entity.

Finally, we examine the extent to which the metadata used to enrich the networks is useful in improving the serendipity of the search results. For this purpose, we conduct this second crowd-sourced evaluation not only on the results extracted from the general, unconstrained YA and WP networks, but also on the constrained setups described at the beginning of Section 7, where we filter the results of the original retrieval based on topic, high and low sentimentality, and high and low readability.

Methodology. As we explained above, our evaluation takes four dimensions into account: *relevance, interestingness for the query, interestingness to the user*, and *learn something new about the query*. Due to the highly subjective nature of these dimensions, we compare the results of our various experimental setups to each other instead of attempting to assign an intrinsic interestingness value to each result. Inspired by Arguello et al [3], we perform pairwise comparisons between all of the result pairs and build a reference result ranking for each dimension. Specifically, given a number of result sets $\{R_1, R_2, ..., R_n\}$ for a query q (obtained from different experimental setups), we take the union of these results, $\boldsymbol{R}$. For each possible pair in $\boldsymbol{R}$ we present the query and the pair to the labeler and we ask which alternative result is preferable, given the four dimensions we take into consideration. Following [3], we allow three choices: "first is better", "second is better", and "both are bad". The items in a pair are randomly positioned as first or second.

Each such task is labeled by three annotators. Incidentally, there are almost no ties, since "don't know" selection is almost never chosen. However, it is prohibitively expensive to label each possible pair, especially if there is little overlap between the result sets. Thus, to estimate the proper rank of a result we sample comparison pairs for each result from all possible ones, and we use a voting methodology to rank them into a reference ranking (note that such rankings may differ across the four dimensions).

The difference between the result ranking in each run and this reference ranking can then be used to gauge the difference between the various runs. We use a rank-based distance metric, Kendall's tau-b, which counts the number of concordant and discordant pairs of items in a list. Given a pair of items $\{x, y\}$ from two lists a and b, the pair is said to be concordant if their ordering matches, that is, $x_i > x_j$ and $y_i > y_j$ or $x_i < x_j$ and $y_i < y_j$. Kendall's tau-b also takes into account ties, where $x_i = x_j$ or $y_i = y_j$, by subtracting the combinations with tied items from the denominator in order to keep the measure in the range of $[-1, 1]$.

Labeling. We used `CrowdFlower.com` to label the sampled $\{query, result_1, result_2\}$ triplets. The query and results were shown along with their Wikipedia pages, and four questions were asked:

1. *Which result is more relevant to the query?*
2. *If someone is interested in the query, would they also be interested in these results?*
3. *Even if you are not interested in the query, are these results interesting to you personally?*
4. *Would you learn anything new about the query?*

To maintain quality, we used settings similar to the ones of the first annotation task. Only the contributors who successfully completed the preliminary training session, providing correct answers to at least four out of six golden questions, were allowed participation. Only the answers to the first question (concerning relevance) were judged for acceptance, being the most objective of the set. Contributors who passed the preliminary training, but then achieved an accuracy on the golden standard lower than 70%, were also excluded. In total, 7,139 tasks were judged (3 annotations for each task), averaging about 13 comparisons for each result. The annotation overlap was 83%, 81%, 76%, and 81% for questions 1, 2, 3, and 4, respectively. It is expected that question 3 would have the lowest agreement, in that it asks specifically the personal opinion of the labeler.

Results. Although highly correlated, the rankings for the four questions are not always the same. When computing

Table 5: Serendipity across different runs: Fraction of unexpected recommendations that are also relevant. In parentheses, the fraction of total recommendations that are both unexpected and relevant.

Baseline	Data	Unconstrained	Topic	High Sent.	Low Sent.	High Read.	Low Read.
Top	WP	0.63 (0.58)	**0.64** (0.59)	0.62 (0.55)	0.46 (0.46)	0.56 (0.53)	0.46 (0.44)
	YA	0.69 (0.63)	0.70 (0.64)	0.67 (0.62)	0.51 (0.49)	**0.71** (0.65)	0.55 (0.54)
	COM	0.70 (0.61)	**0.72** (0.63)	0.70 (0.61)	0.48 (0.46)	0.68 (0.61)	0.52 (0.50)
Top Nwp	WP	0.63 (0.58)	**0.64** (0.60)	0.62 (0.57)	0.46 (0.46)	0.56 (0.54)	0.46 (0.44)
	YA	0.70 (0.64)	**0.71** (0.65)	0.68 (0.64)	0.52 (0.50)	**0.71** (0.66)	0.55 (0.55)
	COM	0.71 (0.64)	**0.73** (0.65)	0.71 (0.64)	0.49 (0.48)	0.68 (0.63)	0.53 (0.51)
Rel	WP	0.64 (0.61)	**0.65** (0.62)	0.64 (0.61)	0.46 (0.46)	0.57 (0.56)	0.47 (0.46)
	YA	0.70 (0.65)	**0.71** (0.66)	0.69 (0.65)	0.52 (0.50)	**0.71** (0.66)	0.56 (0.55)
	COM	0.72 (0.67)	**0.73** (0.68)	0.72 (0.68)	0.49 (0.47)	0.69 (0.65)	0.54 (0.52)
Rel + Top	WP	0.61 (0.54)	**0.63** (0.55)	0.60 (0.52)	0.46 (0.46)	0.55 (0.51)	0.45 (0.42)
	YA	0.68 (0.57)	**0.69** (0.58)	0.66 (0.58)	0.50 (0.47)	**0.69** (0.59)	0.55 (0.54)
	COM	0.68 (0.55)	**0.70** (0.56)	0.68 (0.56)	0.48 (0.45)	0.66 (0.56)	0.52 (0.49)

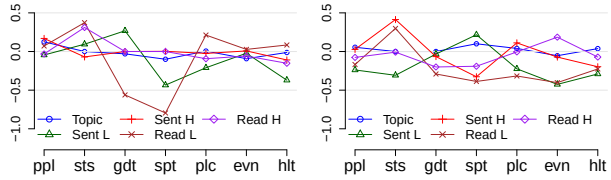
the difference between the proportions of preferences between personal interest (Q3) and relevance (Q1) we find the entities which are interesting, but not necessarily very related to the query. The example with which we started this paper is one of such cases: its rank in relevance ranking is 7 and in personal interestingness 4. Other such gems are a New York Times bestseller *Water for Elephants* for query *Robert Pattinson*, and article on the *History of Ptolemaic Egypt* for *Egypt*. On the opposite end are the entities which are technically relevant but not interesting. These include *Britney Spears* for *Lady Gaga*, *Cairo Conference* for *Egypt*, and *Blu-ray Disc* for *Netflix*.

Table 6 shows Kendall's tau-b between the result sets and the reference ranking. Note that the runs that performed worse on relevance would tend to also have low tau, since the non-relevant entities at the tail of the reference ranking often have the same score (0) and are thus tied for the rank. The table also includes the significance of the difference between the metadata-constrained runs and the unconstrained one using paired t-test (due to larger variance in WP scores, most changes in WP runs are not significant).

For all questions, YA produces results ordered similarly to the reference rank. In fact, for the general, unconstrained runs YA produces better rankings than WP at $p < 0.05$ for all four questions. The correspondence is more pronounced for question 3, which concerns personal interest in the entity. The difference is especially striking, considering a nearly even share of results from both datasets in the top 5 entities of each reference ranking (on average having 2.6 and 2.4 results from WP and YA, respectively).

Constraining search results using topical category improves this similarity (though only for YA at $p < 0.10$). Adding a high-sentimentality constraint also boosts the taus for WP, but the same is not true for YA, where the lack of editorial oversight allows for low-quality highly-emotional posts. For example, the queries with the lowest taus in the high-sentimentality YA run are *Olympic Games*, *José Mourinho*, and *Middle East* – topics involving sports and politics. The queries with the highest taus are those which are less likely to produce spirited discussions, such as *Mount Everest* and *Tsunami*. When constraining to low readability (thus more sophisticated) documents, both precision and similarity to the reference ranking fall dramatically, especially for YA (at $p < 0.05$). This illustrates the difficulty of evaluation of the quality and usefulness of the documents in informal writings, and we leave the exploration of this area for future research.

Figure 3: Change in Kendall's tau-b for interestingness to query (Q2) in the constrained runs (ppl: people, sts: sites, gdt: gadgets, spt: sports, plc: places, evn: events, hlt: health)



(a) Wikipedia (b) Yahoo! Answers

To further understand how metadata constraints affect performance on particular topics, we plot the change in Kendall's tau-b compared to the unconstrained runs in Figure 3. Whereas the low-sentimentality constraint hurts performance for the group *sports* for WP, the opposite is true for YA. Possibly, the already emotionally-intense sports discussions in YA benefit from a selection of less intense documents. For example, the low-sentimentality constraint brings up techniques such as *fast bowling* for *Cricket* and places like *Oriel Park* for *FIFA*, replacing famous sportsmen as top results. It is clear that the effect of metadata constraints differs between the groups, and we leave the topic-specific tuning of these facets for future research.

8. CONCLUSIONS AND FUTURE WORK

This paper investigates the potential of entities extracted from two sources of user-generated content, Wikipedia and Yahoo! Answers, in promoting serendipitous search. Within the context of entity search, we show that both Wikipedia and Yahoo! Answers offer relevant results which are dissimilar to those found through a web search. Also, between the two, the top retrieved entities have often little overlap, suggesting the complementary nature of these two data sources. However, Yahoo! Answers shows to be better at favoring the most interesting entities – both when considering the query and personally to the labelers. Finally, we observe that the effects of metadata constraints vary across datasets and topical categories, suggesting that it is not enough, for instance, to select only emotionally-evocative items in order to catch the user's interest.

Table 6: Similarity (Kendall's tau-b) between result sets and reference ranking (significance of the difference from unconstrained run: * $p < 0.05$, † $p < 0.10$)

	Q1 Relevance			Q2 Int to query			Q3 Personal int			Q4 Learning new		
	WP	YA	Com	WP	YA	Com	WP	YA	Com	WP	YA	Com
Unconstrained	0.162	0.336	0.201	0.162	0.312	0.184	0.139	0.324	0.168	0.167	0.307	0.184
Topic	0.194	0.374†	0.222	0.176	0.343	0.222	0.144	0.359†	0.198	0.164	0.346†	0.203
Low sentimentality	0.042	0.103*	0.133	0.033	0.093*	0.139	0.039	0.113*	0.131	0.039	0.102*	0.101
High sentimentality	0.208	0.303	0.204	0.185	0.289	0.203	0.154	0.310	0.199	0.178	0.289	0.191
Low readability	0.138	0.076*	0.098	0.154	0.079*	0.088	0.153	0.072*	0.060	0.169	0.106*	0.074
High readability	0.118	0.289	0.191	0.120	0.265	0.210	0.087	0.279	0.175	0.118	0.251	0.223

A natural extension to this work is to incorporate the temporal characteristics of recency, trendiness, and novelty into the definition of *goodness*, albeit usefulness or serendipity. Another future direction lies in the exploration of the user's experience during serendipitous search. A prototype is currently in development, aiming to both determine effective ways to incorporate entity search within natural information seeking activities, and to visualize search results to promote serendipity. A study in the context of news search already provides good insights on these two aspects [31].

9. ACKNOWLEDGEMENTS

This work was partially funded by the Linguistically Motivated Semantic Aggregation Engines (LiMoSINe[18]) EU project.

References

[1] E. Amitay, D. Carmel, N. Har'El, S. Ofek-Koifman, A. Soffer, S. Yogev, and N. Golbandi. Social search and discovery using a unified approach. In *HT*, 2009.
[2] P. Andre, J. Teevan, and S. T. Dumais. From x-rays to silly putty via uranus: Serendipity and its role in web search. *ACM SIGCHI*, 2009.
[3] J. Arguello, F. Diaz, J. Callan, and B. Carterette. A methodology for evaluating aggregated search results. In *Advances in information retrieval*, pages 141–152. Springer, 2011.
[4] K. Balog, E. Meij, and M. de Rijke. Entity search: building bridges between two worlds. In *SEMSEARCH*, 2010.
[5] R. Baraglia, G. De Francisci Morales, and C. Lucchese. Document similarity self-join with mapreduce. In *ICDM*, 2010.
[6] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *Proceedings of the 2009 workshop on Web Search Click Data*, WSCD, 2009.
[7] F. Bonchi, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini. Efficient query recommendations in the long tail via center-piece subgraphs. In *SIGIR*, 2012.
[8] I. Bordino, G. De Francisci Morales, I. Weber, and F. Bonchi. From machu picchu to rafting the urubamba river: Anticipating information needs via the entity-query graph. In *WSDM*, 2013.
[9] A. Bozzon, M. Brambilla, S. Ceri, and P. Fraternali. Liquid query: multi-domain exploratory search on the web. In *WWW*, 2010.
[10] K. Chakrabarti, V. Ganti, J. Han, and D. Xin. Ranking objects based on relationships. In *SIGMOD*, 2006.
[11] T. Cheng, X. Yan, and K. C.-C. Chang. Entityrank: searching entities directly and holistically. In *VLDB*, 2007.
[12] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR*, 2007.
[13] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *PODS*, 2004.
[14] R. Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32(3):p221 – 233, June 1948.
[15] M. Ge, C. Delgado-Battenfeld, and D. Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *RecSys '10*, pages 257–260, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-906-0.
[16] C. Hauff and G.-J. Houben. Serendipitous browsing: Stumbling through wikipedia. *Searching 4 Fun*, 2012.

[18] www.limosine-project.eu

[17] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *EMNLP*, 2011.
[18] L. Iaquinta, M. De Gemmis, P. Lops, G. Semeraro, M. Filannino, and P. Molino. Introducing serendipity in a content-based recommender system. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*, pages 168–173. IEEE, 2008.
[19] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD*, 2002.
[20] G. Jeh and J. Widom. Scaling personalized web search. In *WWW*, 2003.
[21] J. Karlgren, M. Sahlgren, F. Olsson, F. Espinoza, and O. Hamfors. Usefulness of sentiment analysis. In *ECIR*. 2012.
[22] J. Y. Kim, K. Collins-Thompson, P. N. Bennett, and S. T. Dumais. Characterizing web content, user interests, and search behavior by reading level and topic. In *WSDM*, 2012.
[23] H. Knäusl. Searching wikipedia: learning the why, the how, and the role played by emotion. *Searching 4 Fun*, 2012.
[24] Y. Koren, S. C. North, and C. Volinsky. Measuring and extracting proximity in networks. In *KDD*, 2006.
[25] O. Kucuktunc, B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu. A large-scale sentiment analysis for yahoo! answers. In *WSDM*, 2012.
[26] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in web text. In *KDD*, 2009.
[27] Y. Liu and E. Agichtein. On the evolution of the yahoo! answers qa community. In *SIGIR*, 2008.
[28] C. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*. CUP.
[29] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, 2007.
[30] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *CIKM*, 2008.
[31] Y. Moshfeghi, M. Matthews, R. Blanco, and J. M. Jose. Influence of timeline and named-entity components on user engagement. In *ECIR*, 2013.
[32] H. O'Brien. Exploring user engagement in online news interactions. *ASIST*, 48(1):1–10, 2011.
[33] B. O'Connor, M. Krieger, and D. Ahn. Tweetmotif: Exploratory search and topic summarization for twitter. *ICWSM*, 2010.
[34] D. Paranjpe. Learning document aboutness from implicit user feedback and document structure. In *CIKM*, 2009.
[35] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11), 1975.
[36] F. Sebastiani. Text categorization. In *Encyclopedia of Database Technologies and Applications*. 2005.
[37] D. Shahaf, C. Guestrin, and E. Horvitz. Trains of thought: Generating information maps. In *WWW*, 2012.
[38] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.*, 63 (1):163–173, Jan. 2012. ISSN 1532-2882.
[39] E. G. Toms. Serendipitous information retrieval. In *DELOS*, 2000.
[40] H. Tong and C. Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *KDD'06*, pages 404–413. ACM, 2006.
[41] S. Yogev, H. Roitman, D. Carmel, and N. Zwerdling. Towards expressive exploratory search over entity-relationship data. In *WWW Companion*, 2012.
[42] Y. Zhang, D. Séaghdha, D. Quercia, and T. Jambor. Auralist: introducing serendipity into music recommendation. In *WSDM*, 2012.
[43] Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney. Resolving surface forms to Wikipedia topics. In *COLING*, 2010.