Leveraging Knowledge Bases for Contextual Entity Exploration

Joonseok Lee*
Google Inc.
Mountain View, CA, USA
joonseok@google.com

Ariel Fuxman†
Google Inc.
Mountain View, CA, USA
afuxman@google.com

Bo Zhao†
LinkedIn
Mountain View, CA, USA
bo.zhao.uiuc@gmail.com

Yuanhua Lv
Microsoft Research
Redmond, WA, USA
yuanhual@microsoft.com

ABSTRACT

Users today are constantly switching back and forth from applications where they consume or create content (such as e-books and productivity suites like Microsoft Office and Google Docs) to search engines where they satisfy their information needs. Unfortunately, though, this leads to a suboptimal user experience as the search engine lacks any knowledge about the content that the user is authoring or consuming in the application. As a result, productivity suites are starting to incorporate features that let the user "explore while they work".

Existing work in the literature that can be applied to this problem takes a standard bag-of-words information retrieval approach, which consists of automatically creating a query that includes not only the target phrase or entity chosen by the user but also relevant terms from the context. While these approaches have been successful, they are inherently limited to returning results (documents) that have a syntactic match with the keywords in the query.

We argue that the limitations of these approaches can be overcome by leveraging semantic signals from a knowledge graph built from knowledge bases such as Wikipedia. We present a system called Lewis for retrieving contextually relevant entity results leveraging a knowledge graph, and perform a large scale crowdsourcing experiment in the context of an e-reader scenario, which shows that Lewis can outperform the state-of-the-art contextual entity recommendation systems by more than 20% in terms of the MAP score.

Categories and Subject Descriptors

H.4 [**Information systems applications**]: Data mining

*This work was done during an internship at Microsoft.

†This work was done while working at Microsoft.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2788564.

Keywords

Entity recommendation, Context, Semantic, Knowledge base, Context-Selection Betweenness

1. INTRODUCTION

Users today are constantly switching back and forth from applications where they consume or create content (such as e-books and productivity suites like Google Docs and Microsoft Office) to search engines where they satisfy their information needs (such as Bing or Google). Unfortunately, though, this leads to a suboptimal user experience as the search engine lacks any knowledge about the content that the user is authoring or consuming in the application [11, 19, 13].

How can we empower users to satisfy their information needs directly within the applications where they consume content? A significant step in this direction is enabling users to interact with anything on the document that they are working on, directly within the productivity application, and recommending results that are contextually relevant to the elements they are interacting with. Productivity suites are starting to incorporate features that realize this scenario, such as the "Insights for Office" feature in Microsoft Word Online.

As an example, consider a user reading on an e-reader the document shown in Figure 1, which describes the Capture of Fort Ticonderoga, an important event in American history. At some point, she finds a mention to a historical figure called "Silas Deane" and decides that she would like to learn more about him. Just sending the query "silas deane" to any of the major commercial search engines returns results such as "Silas Dean High School" which are unrelated to the historical context of the document. A much more compelling user experience is the one shown in Figure 1, where the user has tapped on the phrase "Silas Deane" and is shown contextually relevant articles such as "Revolutionary War", where she can learn about Silas Deane's overall involvement in the American Revolutionary War, and "Benjamin Franklin", where she can learn that Deane and Franklin were the first diplomats in American history, and they were sent together to France as commissioners from the Continental Congress.

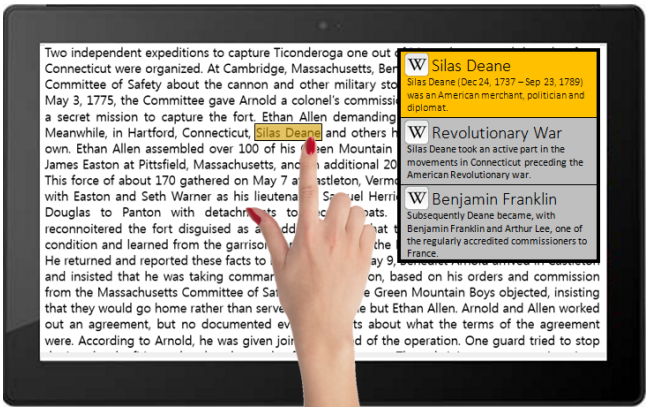

Figure 1: An example of contextual exploration.

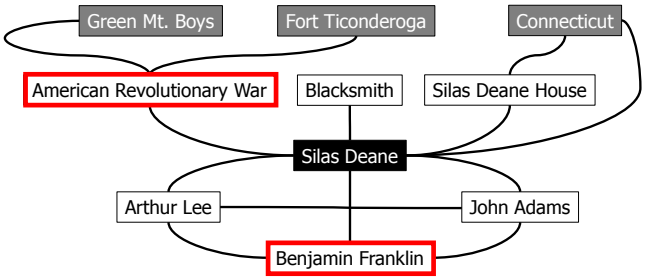

Figure 2: A portion of the focused subgraph for our running example. (Black for the user selection node, gray for context nodes, and white for all other nodes.)

Previous efforts in the literature have made significant progress towards realizing this scenario, including systems for *contextual search* [11, 19] and *contextual insights* [13]. These systems take a standard bag-of-word information retrieval approach to the problem, which consists of automatically creating a query that includes not only the target phrase or entity chosen by the user but also relevant terms from the context. More broadly, these approaches are related to relevance feedback in information retrieval [35], where a context (in the case of relevance feedback, the results of a query; in our scenario, the document context) is used to refine an initial query (in our case, the phrase chosen by the user).

While these approaches have been successful, they are inherently limited to returning results (documents) that have a syntactic match with the keywords in the query. For example, the Wikipedia articles for "Revolutionary War" and "Benjamin Franklin" have just a single passing mention to Silas Deane and are thus unlikely to be retrieved by a query that contains the terms "silas deane". To tackle this problem, in this paper we argue that such results can be obtained by more directly modeling the semantic connections between the target concept and the entities in the context where it appears.

To illustrate our approach, consider the graph shown in Figure 2 (henceforth called *knowledge graph*). The black node corresponds to the entity chosen by the user (Silas Deane), the gray nodes correspond to entities mentioned in the context (Green Mountain Boys, Fort Ticonderoga, Connecticut). The edges correspond to hyperlinks in the Wikipedia articles. As we can see, the node for "Revolutionary War" acts as a bridge between Silas Deane and the context concepts Green Mountain Boys (the militia that captured Fort Ticonderoga) and Fort Ticonderoga. Our approach leverages precisely this type of semantic connections to retrieve contextually relevant results.

The contributions of this paper include:

- A framework for leveraging semantic signals from a knowledge graph for the problem of retrieving contextually relevant entity results.
- A system called *Lewis* for retrieving contextually relevant entities leveraging a knowledge graph built from Wikipedia hyperlinks.
- An algorithm for retrieving contextually relevant entities.
- A large-scale evaluation of the approach in the context of a real-word e-reader application. The results show an improvement of up to 20.8% in terms of the MAP scores with respect to the state-of-the-art methods for contextual insights and pseudo-relevance feedback. We also present a detailed ablation study that shows the importance of the different components of the *Lewis* system.

The rest of this paper is organized as follows. We formally define the contextual entity exploration problem in the next section, followed by a detailed description of our proposed method in Section 3. We evaluate our method in Section 4. Lastly, we review related problems and previous work in Section 5, and provide concluding remarks in Section 6.

2. CONTEXTUAL ENTITY EXPLORATION PROBLEM

The input to the contextual entity exploration problem consists of a *user selection*: the span of text that the user highlights with the mouse or taps with the finger, which implicitly determines the entity she would like to gain insights about (e.g., "Silas Deane"); a *context*, consisting of the content that the user is consuming or authoring; a *knowledge base*, that consists of entities that are candidates to be recommended; and a *knowledge graph*, whose nodes are entities from the knowledge base; and an *text-to-entity* mapping that given some text from the user selection or context produces an entity from the knowledge base.

The *contextual entity exploration problem* is then formally defined as follows:

DEFINITION 1. *Given a quintuple (s, C, B, G, γ), where s is a user selection, C is some text context, B is a knowledge base, G is a undirected graph whose nodes are entities in B, and γ is a text-to-entity mapping; the objective of the* contextual entity exploration problem *is to produce a set of entities O such that $O \subseteq B$ and every entity in O is relevant to s in the context of C.*

In this work, we will use Wikipedia as our knowledge base B and the hyperlink structure of Wikipedia as the edges of the knowledge graph. In particular, G will be an undirected graph $G = (B, E)$ where there is an edge (x, y) in E if there

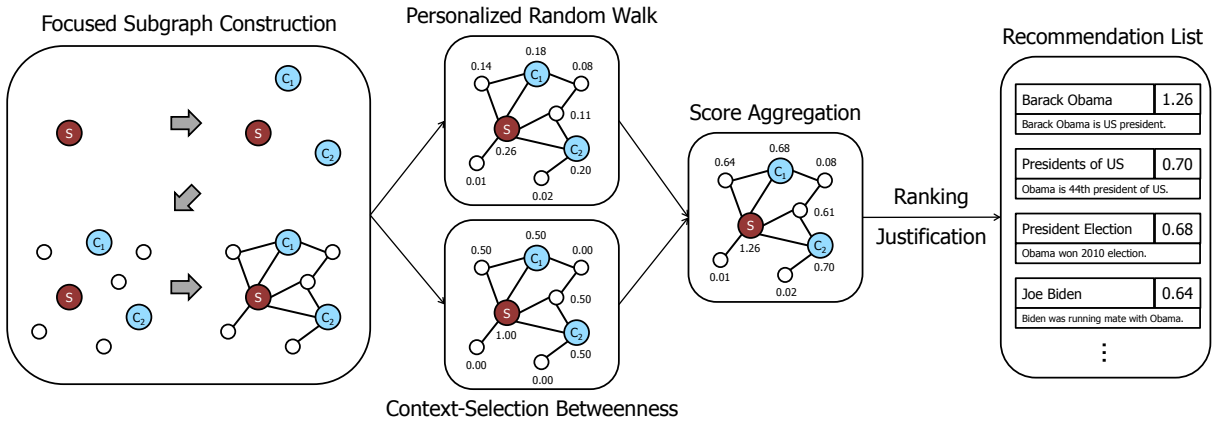

Figure 3: Overall flow of the proposed approach.

is a link to entity y on the Wikipedia page for entity x or vice versa. Notice, however, that the techniques presented in this paper are independent of the actual method used to construct the knowledge graph.

3. OUR APPROACH

In this section, we explain our approach to contextual entity exploration using a knowledge graph. Figure 3 provides an overview of the proposed system. We start by building a focused subgraph from the given knowledge graph for each problem instance, followed by scoring participating nodes in two ways, namely context-selection betweenness and personalized random walk. Ranking the aggregated scores generates a recommendation list, with a human readable justification.

3.1 Focused Subgraph

The first step of the algorithm consists of mapping the user selection s and the context C to nodes in the knowledge graph using the text-to-entity mapping γ. Notice that this mapping is given as input to the contextual entity exploration problem; it can be any off-the-shelf entity linking system (e.g., [9])[1].

Continuing our running example of Figure 1, a mapping γ would map the user selection "Silas Deane" to the entity for Silas Deane in Wikipedia[2]; and extract from the document entities related to the surface forms that appear therein, e.g., "Green Mountain Boys"[3] and "Fort Ticonderoga"[4].

The next step consists of creating a subgraph of the knowledge graph that contains candidate entities to be recommended as contextually relevant results, which we call *focused subgraph*. Its nodes consist of the union of two sets V' and V''. V' is the set of entities obtained by applying the mapping γ in the previous step (i.e., the entities associated to the user selection and context); and V'' is the set of entities reachable from nodes of V' in the knowledge graph G through a path of length one. The edges of the focused subgraph can be obtained by adding the edges induced by $V' \cup V''$ in G, or by any other suitable heuristic.[5] Figure 2 demonstrates an example of the focused subgraph for our running example.

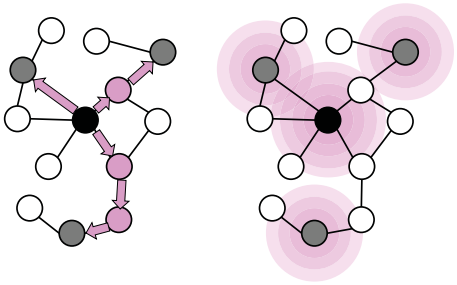

Figure 4: Illustration of retrievable nodes with context-selection betweenness (*left*) and personalized random walk (*right*). (The black node in the center represents the user selection node, and three gray nodes are context pages.)

Once the focused subgraph is constructed, we proceed to scoring each candidate entity in the focused subgraph by capturing the semantic connection among the candidates, the user selection, and the context nodes. We explain the scoring methods in detail next.

3.2 Scoring Methods

We now present the two scoring methods that we use, and how we combine them to produce a final contextual relevance score.

3.2.1 Context-Selection Betweenness

This method captures to what extent a given candidate node serves as a bridge between the user selection node and the context nodes. For example, in Figure 2, "Revolutionary War" gets a higher score than "Blacksmith" because remov-

[1]For our experiments, we use an in-house entity linking system. We also conduct experiments where the mapping γ is actually an oracle: i.e., a human manually provides a perfect mapping.

[2]http://en.wikipedia.org/wiki/Silas_Deane

[3]http://en.wikipedia.org/wiki/Green_Mountain_Boys

[4]http://en.wikipedia.org/wiki/Fort_Ticonderoga

[5]The heuristic that we employ in the Lewis system consists of adding all the edges that involve at least one context or user selection node; and the edges (x, y) such that either x has inlinks from the user selection node s and y in G (s and y are common parent of x), or x has outlinks to the user selection node s and y in G (s and y are common children of x).

ing the former disconnects "Silas Deane" from two context nodes, whereas removing the latter does not disconnect the selection node from any context node. This makes intuitive sense because while the Revolutionary War is very relevant to Silas Deane and the context of the document, the entity "Blacksmith" is irrelevant to the context (it is just mentioned on the Wikipedia page for Silas Deane because his father was a blacksmith).

We capture this intuition with a measure that we call *context-selection betweenness* (CSB). The measure is inspired by the notion of betweenness centrality [12], which measures how many shortest paths go through a given node v. It is defined as follows:

$$BC(v) = \sum_{v \neq i \neq j} \frac{\sigma_{i,j}(v)}{\sigma_{i,j}}, \quad (1)$$

where $\sigma_{i,j}$ is the total number of shortest paths from node i to j, and $\sigma_{i,j}(v)$ is the number of such paths that pass through node v.

For the contextual entity exploration problem, however, the original definition of betweenness centrality does not suffice. First, in the contextual entity exploration problem, relevance is defined with respect to the context. Thus, the measure should consider only paths connecting the user selection node and the context nodes, not every path connecting two nodes. This can be done by a straight-forward modification of the original definition, by simply by summing over all (s, c) pairs for all $c \in C$, instead of all possible pairs of nodes in the focused graph.

Another reason that we need to modify betweenness centrality is that not all context terms are equally relevant to the user selection. Thus, we need a weighted version of betweenness centrality. To compute this weight, we use Normalized Wikipedia Distance (NWD) [28], a measure of semantic distance of two nodes on graph that is widely used in the entity linking literature. The NWD between two nodes u and v is defined as

$$NWD(u, v) = \frac{\log(\max(|I_u|, |I_v|)) - \log |I_u \cap I_v|}{\log |V| - \log(\min(|I_u|, |I_v|))}, \quad (2)$$

where I_x is the set of incoming edges to the node x, and V is the set of all nodes in Wikipedia. In our modified measure, each path from user selection node s to a context node c is weighted by $\max(1/(\theta - NWD(s, c)), 0)$ with some threshold θ. (In our case, $\theta = 0.5$.) In this way, a context term more relevant to the target mention is more emphasized.

Putting it all together, we define *context-selection betweenness* of a node v as follows:

Definition 2. *Context-Selection Betweenness of a node v is*

$$CSB(v) = \frac{1}{Z} \sum_{c: v \in sp(s,c)} \frac{w(s, c)}{k \cdot l(s, c)}, \quad (3)$$

where $w(s, c) = \max(\theta - NWD(s, c), 0)$, $l(p, c)$ is the length of shortest path between user selection node s and context node c, $sp(s, c)$ is a set of all shortest paths between s and c, $Z = \sum_{c \in C} \frac{w(s,c)}{l(s,c)}$, and k is the number of different shortest paths between s and c.

Intuitively speaking, a higher CSB score means the node is playing a more important role connecting the user selection node s and other context nodes c, which are more relevant to p through shorter and more unique paths.

Figure 4 (left) illustrates how CSB works. We have three context (gray) nodes in this graph. For each user selection - context $(s - c)$ path, marked with pink arrows, we assign scores by (3) to all participating nodes on it. As a result, we found three pink nodes including A with nonzero score. All other white nodes are 0, as they are not on any $s - c$ path.

3.2.2 Personalized Random Walk

We also consider a measure of the relevance of a node to the user selection. This measure does not factor in the contributions of context nodes directly, but it does so indirectly since the focused graph is built from context nodes in the knowledge graph. In Section 4, we will show that while this measure does not suffice by itself to obtain an appropriate relevance score, it is quite effective when used in conjunction with the context-selection betweenness measure described above.

We compute this measure by computing a personalized random walk [18]. Intuitively, the random walk is simulating the behavior of a user reading articles in Wikipedia as follows. We assume that a user starts reading the page most directly relevant to the user selection (the page for "Silas Deane" in our example) and then follow an interesting link from that page. She continues surfing articles in this way, until at some point she comes back to the article of the user selection.

Random walk [30] computes the stationary probability that a user would stay in the page. Personalized random walk [18] is a generalization of random walk in that it introduces a set of default pages which the user can jump from anywhere on the web with certain probability. It is proven to always have a unique solution with stationary probability for each page. [18] There are numerous previous works using personalized random walk for graph-based data mining. The most related to ours is WikiWalk [41], which used a general jump probability vector (possibly with different probability to different pages) for measuring similarity of two texts (without a notion of "user selection"). Note that we call this random walk "personalized" only because it is the name known in literature, not because we personalize to the interests of an individual user.

To compute the random walk, we use a vector for all nodes, containing random jump probability to each node. We assign $0 < x_s < 1$ for the user selection node s and $x_c/|C|$ where $0 \leq x_c \leq x_s$ for each context node $c \in C$. All the other nodes are assigned zero probability of random jump. Thus, users can come back to the user selection node during surfing with some probability x_s, as well as to some context page with smaller probability $x_c/|C|$, and restart navigation from there.

Figure 4 (right) illustrates an example of probability distribution from personalized random walk with $x_s > x_c > 0$. The user tends to stay on the user selection node with highest probability, followed by context nodes. Also, nodes close from the user selection and context nodes have slightly higher probability than nodes far from them. In other words, personalized random walk retrieves semantically relevant pages from the query and context terms by assigning higher probability (score) to closely and densely connected nodes from the user selection and context nodes.

3.2.3 Score Aggregation

At this point, we have two scores for each node v: a context-selection betweenness score $CSB(v)$, and a random

walk score $RW(v)$. We now explain how to combine them to obtain the final relevance score.

The random walk scores of a node are probability scores and thus sum up to 1. Thus, the expected value of $RW(v)$ gets smaller when we have more nodes in the graph. To counter the effect of graph size, we consider $|V|RW(v)$ instead of $RW(v)$ itself, where V is the set of nodes in the focused graph. As the expected value of $|V|RW(v)$ is always 1 because it sums to $|V|$, we can interpret this score as how many times the node is preferred to visit compared to expectation. If $|V|RW(v) = 3$, for example, we interpret the page v is 3 times more recommendable than others.

Context-selection betweenness, on the other hand, is normalized by the sum over all context nodes. Thus, the $CSB(v)$ score for each node tends to be inversely proportional to the number of context nodes. To counter this effect, we again consider $|C|CSB(v)$ instead of $CSB(v)$, where C is the set of context nodes. As the highest score of $|C|CSB(v)$ is $|C|$, each $s-c$ path distributes 1 to all participating nodes. Thus, we can interpret $|C|CSB(v)$ score as the expected number of shortest paths from user selection s to any context node visiting v in the meanwhile.

We aggregate these scores reflecting their relative importance. First, it is natural to trust context-selection betweenness score more when we have more context terms at hand. On the other hand, we trust context-selection betweenness less when we have a relatively large number of nodes in our focused graph compared to the number of context nodes $|C|$, as this may imply that either nodes in outside the context may not overlap so much (that is, the context is not topically coherent) or user selection and context nodes have large number of connected nodes (so they are general terms, e.g, *Water* or *Human*). In either case, therefore, the importance of $|C|CSB(v)$ should be proportional to the ratio of $|C|$ to $|V|$. With a scaling factor α, we finally propose the following aggregation equation to compute final score for node v:

DEFINITION 3. *Relevance score of a node v is given by*

$$Relevance(v) = |V|RW(v) + \alpha\frac{|C|}{|V|}|C|CSB(v). \quad (4)$$

We sort this score for each entity in decreasing order, and recommend the top-k entities. We recommend nodes v satisfying $|V|RW(v) > 1$ only. That is, we do not recommend pages with lower random walk score even than its expectation. This is to remove some general (so not recommendable) context terms having very high $|C|CSB(v)$ due to a cluster of context pages on the focused subgraph.

4. EVALUATION

In this section, we evaluate our approach and compare it with several baselines using crowd-sourced data in the context of a real-world e-reader application.

4.1 Experimental Setting

Dataset

We employed snapshot of English Wikipedia from January 2nd, 2014 as our knowledge base, considering all pages from namespaces *main* and *category*. We further performed some preprocessing: removing stop words, consolidating redirections[6], and removing disambiguation pages[7] since they connect ambiguous entities which are not quite related with each other. (This is different from category or list pages, which contain semantically relevant pages.)

We performed experiments in the context of an e-reader application, as illustrated in Figure 1. To create suitable test data, we employed a corpus consisting of all English textbooks from the Wikibooks site[8]. The corpus consists of 2,600 textbooks that cover a broad spectrum of topics, such as engineering, humanities, health sciences, and social sciences. We sampled 900 paragraphs from this corpus, and for each paragraph we asked 100 crowd workers to select phrases for which they would like to learn more. Then, we performed weighted random sampling from the user-selected phrases to get 500 test cases (pairs of user selections and contexts). For each test case, we pooled the top 8 results from our system as well as several baselines. For each result in the pool, we showed the original user selection and context to 10 crowd workers and ask them if they thought the recommended page is good in the context. We applied some simple heuristics to remove spam labels, and used majority voting to get the final label.

Model Parameters and Metrics

We considered 100 words before and after the user selection as context for all compared methods. For the personalized random walk, we used $x_s = 0.05$ (random jump probability to the perfect node), $x_c = 0$ (random jump probability to any context node)[9], and iterated up to 50 times. For context-selection betweenness, we used $\theta = 0.5$. We compared Lewis to baselines using Mean Average Precision (MAP) to take both precision and recall into account with a single metric.

User Interface and Justification

Figure 5 shows our user interface for crowd workers to evaluate our recommendation. On the left side, workers can see the original context. The user selection is marked as light green box. On the right side, it shows an entity recommended for the user selection. In order to let the user understand whether the entity is appropriate, we show the Wikipedia page of the entity. The workers are asked to answer how relevant the entity is to the user selection and the given context on the left-bottom of the page. We gave three options: 1) This article is what I'd expect to see if I highlighted the text on the left, 2) This article is not what I'd expected, but I see a connection between the highlighted text and the article, and 3) This article is not what I'd expected, and I don't see a connection between the highlighted text and the article. We regarded 1) and 2) as relevant, and 3) as not relevant.

During this evaluation, we faced the following challenge. As our corpora consist of various topics including literature, history, science, or engineering, it is rather difficult to find a worker sufficiently knowledgeable in all of these areas. Fur-

[6]E.g, "MIT" and "Massachusetts Institute of Technology" refer to the same page. Users might link to this page using either of them.

[7]E.g, "Apple (disambiguation)" page contains links to both the fruit apple and the IT company Apple Inc.

[8]`http://en.wikibooks.org`

[9]We tried some $x_c > 0$, but observed no significant difference.

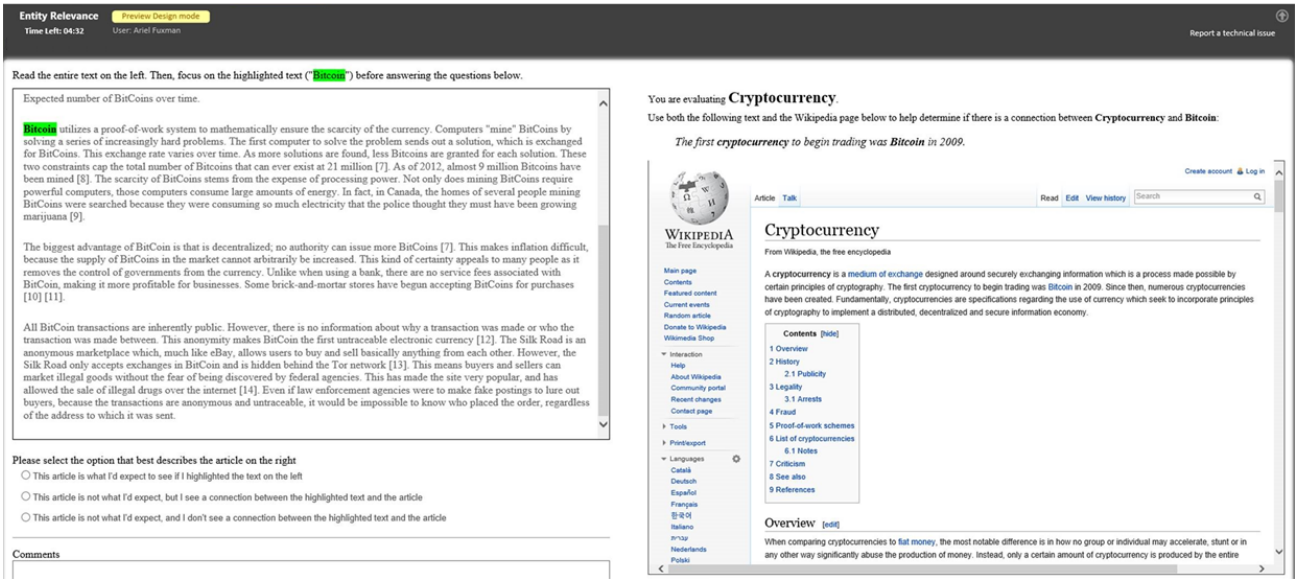

Figure 5: The web page we used for crowd-sourced evaluation.

User selection	Recommend page	Explanation
Wang Mang	Xin Dynasty	**Wang Mang** was a Han Dynasty official who seized the throne from the Liu family and founded the **Xin Dynasty**, ruling AD 9-23.
Malacca	Malaysia	**Malacca** (Melaka, dubbed "The Historic State") is the third smallest **Malaysia**n state after Perlis and Penang.
Diolkos	Corinth Canal	Sections of the **Diolkos** have been destroyed by the 19th-century **Corinth Canal**.
Kanji	Chinese characters	**Kanji** are the adopted logographic **Chinese characters** that are used in the modern Japanese writing system along with hiragana and katakana.
Throttle	Automobile	In a motor vehicle the control used by the driver to regulate power is sometimes called the **throttle** pedal or accelerator.

Table 1: Examples of recommendation justification.

thermore, crowd workers tend to answer without close inspection but just relying on their background knowledge.

To resolve this issue, we added a justification sentence in the user interface (on the top-right of the page). There is recent work in the area of justification of recommendations [16, 37, 43, 39, 8], and in our case we used a simple, yet effective, heuristic. From either the perfect page or the recommended page, we chose one sentence which satisfies one of the following conditions: 1) a sentence containing both (the perfect and the recommended) pages' titles with hyperlink, 2) a sentence containing both (the perfect and the recommended) pages' titles without hyperlink, 3) a sentence containing the other page's title with hyperlink, 4) a sentence containing the other page's title without hyperlink. In this way, we prefer a sentence which can explain relationship between the user selection and the recommended page. If we find more than one sentence in the same category, we chose the first occurence, because general explanation tends to come first in Wikipedia articles. If we can find no sentence satisfying any of those four conditions, we choose the first sentence of the recommended page. Table 1 illustrates several examples of our justifications.

We provided the same justifications for all baselines as well as our method to be fair.

4.2 Results and Discussion

Comparison against Baselines

We considered four baselines. The first baseline consists of simply sending the user selection to a commercial search engine, without using any context. The second baseline consists of the widely used semantic relatedness measure Normalized Wikipedia Distance (NWD) [28]. In this baseline, we take the user selection node in the graph and compute NWD with respect to all other nodes in the focused subgraph, and then return the entities with the top-k score. Finally, we consider two approaches that are representative of contextual entity exploration using bag-of-words IR techniques: the Leibniz system [13], and an algorithm based on positional relevance model (PRM) [25, 14], which is a state-of-the-art pseudo-relevance feedback algorithm. Notice that both NWD and Lewis need to perform entity linking to get the user selection entity. In our experiments, we placed the entity proposed by an in-house entity linking system at the top position for NWD and Lewis, which is for the benefit of fair comparison with the search engine and Leibniz, since their top results are generally entity linking results.

Table 2 compares performance of Lewis against the baselines in terms of MAP@8. We can see that Lewis outperforms all the baselines, with a MAP@8 score of 0.291. In particular, it outperforms the state-of-the-art contextual entity recommendation systems: the PRM pseudo-relevance feedback system, which achieves a MAP@8 score of 0.278; and Leibniz, which achieves a MAP@8 score of 0.262. The lowest MAP score of 0.244 corresponds to using NWD as a measure for entity recommendation.

The scores above correspond to the case in which we output up to eight entities. We also analyzed the case when we set a threshold in the maximum number of results. That is, we computed MAP@k, for varying values of k. The results

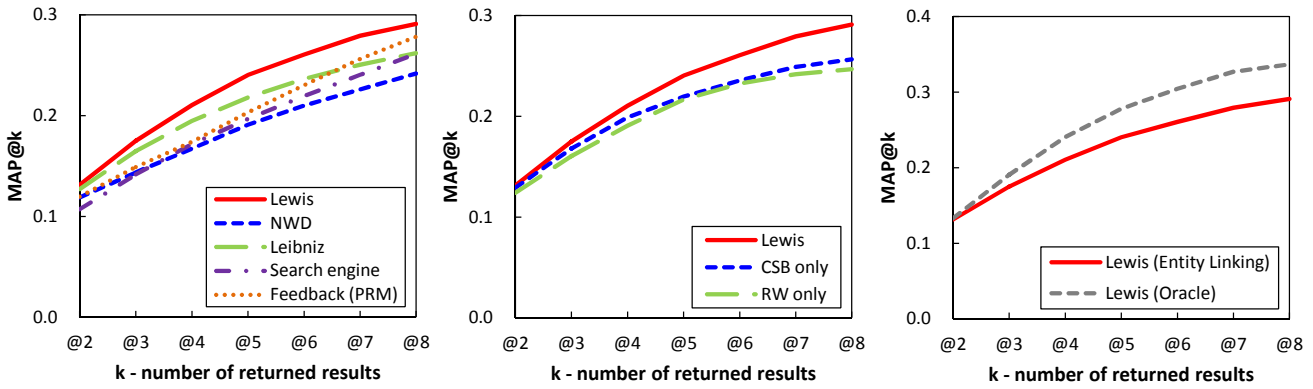

Figure 6: MAP@k for Lewis with baselines (left), for ablation study (middle), and with the user selection entity chosen by entity linking and by human (right).

Method	MAP@8
NWD	0.244
Search Engine	0.261
PRM Relevance Feedback	0.278
Leibniz	0.262
Lewis	**0.291**

Table 2: Comparison of Lewis against baselines.

Method	MAP@8
Random Walk only	0.246
Context-selection betweeness only	0.256
Lewis	**0.291**

Table 3: Ablation study results.

are seen in Figure 6 (left). As we can see, Lewis consistently outperforms all baselines at every value of k.

Ablation Study

We also performed an ablation study where we consider each of the two scoring methods of Lewis in isolation. The results for MAP@8 are shown in Table 3. We can see that the full Lewis system outperforms its ablations. The results for MAP@k for varying values of k are shown in Figure 6 (middle). We can also see that Lewis consistently outperforms the ablations at every value of k. In addition, we see that context-selection betweenness outperforms the random walk method. This means that user selection and context should be considered as first class citizens, as we advocate in this paper.

Effect of Entity Linking

Recall that in the construction of the focused graph, we use a mapping γ to map the user selection to an entity node. All the results given so far are based on the use of an in-house entity linker. But this raises the following question: what is the dependency of Lewis on the actual entity linking method? To address this question, we consider an experiment where the mapping γ is given by an oracle. That is, a human manually produced the correct entity linking result for the user selection. The Lewis system with an entity linking oracle gets a MAP@8 score of 0.336. In contrast, Lewis achieves a MAP@8 score 0.291 when we use our in-house entity linker. This means that if we used a different entity linker system, the MAP@8 score could be improved by at most 17%. Potential MAP improvement with different k is shown in Figure 6 (right).

Rank	Title of Retrieved Page	CSB	RW	Total
1	Silas Deane	402.26	6442.53	6844.78
2	American Revolutionary War	49.58	82.42	131.99
3	Benjamin Franklin	2.46	128.17	130.63
4	Arthur Lee (diplomat)	0.00	107.99	107.99
5	Thomas Jefferson	2.46	98.82	101.28
6	Continental Congress	19.58	74.02	93.60
7	Continental Army	30.25	61.45	89.14
8	Capture of Fort Ticonderoga	49.58	39.56	89.14

Table 4: Score decomposition with *Silas Deane* example.

Overlap of Results

We also investigated how many entities Lewis and Leibniz return in common. Interestingly, only a small portion of the recommended entities overlap. Specifically, Lewis recommended 3,790 entities for 500 test examples (up to 8 entities for each) and Leibniz did so for 3,032 entities. Among these, only 580 entities (9.3%) were in both sets. Furthermore, if we exclude the user selection entity obtained via the entity linker, the number of overlapping entities drops to 320 (5.6%). This observation leads to a conclusion that IR-based and knowledge-graph based systems tend to retrieve qualitatively different types of entities, and thus the combination of such system is a promising direction of future work.

Graph Construction and Running Time

The focused subgraph we created for each problem instance contained 16,041 nodes and 118,380 edges in average. For each instance, in average, it took 0.3 seconds for constructing focused subgraph. It took 3.3 seconds in average for computing context-selection betweenness score, and 3.2 seconds in average for random walk score. (This excludes all preprocessing steps which were done offline, taking about 30 minutes.) The largest focused subgraph had 155,711 nodes and 1,617,403 edges, taking 0.6 seconds for constructing subgraph, and 32 seconds for each scoring method.

Anecdotal Examples

We present a couple of anecdotal examples. We start with running example of Figure 1, where the user selection is *Silas*

CSB only	RW only	NWD baseline	Leibniz
Silas Deane	**Silas Deane**	**Silas Deane**	**Silas Deane**
Ethan Allen	**Benjamin Franklin**	**Arthur Lee (diplomat)**	Silas Deane House
Noah Phelps	**Arthur Lee (diplomat)**	Hugh Campbell Wallace	Connecticut Route 99
Benedict Arnold	**John Adams**	Somerville Pinkney Tuck	Silas Dean House
Fort Ticonderoga	Thomas Jefferson	Evan G. Galbraith	
American Revolutionary War	**James Monroe**	Howard H. Leach	
Capture of Fort Ticonderoga	William Short	**Jesse I. Straus**	
Green Mountain Boys	Gouverneur Morris	Arthur K. Watson	

Table 5: Retrieved results with Lewis and baselines. Bold faces mean relevant results.

Deane. The original text is as follows, with the user selection *Silas Deane* marked with a surrounding box.

Two independent expeditions to capture Ticonderoga – one out of Massachusetts and the other from Connecticut – were organized. At Cambridge, Massachusetts, Benedict Arnold told the Massachusetts Committee of Safety about the cannon and other military stores at the lightly defended fort. On May 3, 1775, the Committee gave Arnold a colonel's commission and authorized him to command a secret mission to capture the fort. Ethan Allen demanding the surrender of Fort Ticonderoga Meanwhile, in Hartford, Connecticut, Silas Deane and others had organized an expedition of their own. Ethan Allen assembled over 100 of his Green Mountain Boys, about 50 men were raised by James Easton at Pittsfield, Massachusetts, and an additional 20 men from Connecticut volunteered. This force of about 170 gathered on May 7 at Castleton, Vermont.

- Silas Deane
- American Revolutionary War
- Benjamin Franklin
- Arthur Lee (diplomat)
- Thomas Jefferson
- Continental Congress
- Continental Army
- Capture of Fort Ticonderoga

Lewis retrieved contextually relevant results such as *American Revolutionary War* and *Capture of Fort Ticonderoga*. We see that part of these results were retrieved by the random walk component, while for some others context-selection betweenness contributed more. Table 4 shows decomposed scores for CSB and RW contributions. For example, context-selection betweenness plays an important role to retrieve *American Revolutionary War* and *Capture of Fort Ticonderoga*, because they are semantically connecting user selection *Silas Deane* and context phrase *Fort Ticonderoga*. It also retrieves results for *Benjamin Franklin*, *Arthur Lee*, and *Thomas Jefferson*, who happened to be the other three people who, together with Silas Deane, can be considered to be the first diplomats of the United States. We see that the random walk component indeed plays more important role for retrieving these pages from Table 4. They are recommended due to their close relation with *Silas Deane*, despite they are not directly related to the context above.

Table 5 compares the output of the *Silas Deane* example from baselines. As we have seen above, CSB only retrieves contextually relevant pages. RW only and NWD baseline mostly retrieve pages for close figures of Silas Deane. Many of them are marked as unrelevant, because they are not related to the context in spite of their relatedness to Silas Deane. Interestingly, Leibniz retrieved only 4 results for this example, where none of them except for *Silas Deane* is relevant to the given context. On the other hand, the entire Lewis system retrieves more relevant results to the context and the user selection by combining CSB and RW component.

The other example is an article about pronunciation and location of stress on the Greek word *Ulysses*. The user selection is *Oxford English Dictionary*:

In 1895, when Joyce was in his third year at Belvedere College, he chose Ulysses as his subject for an essay entitled "My Favourite Hero". In English Ulysses is sometimes stressed on the second syllable, and this is the pronunciation required in most verse translations of the Homeric epics. Joyce, however, always referred to his novel as YOOL-i-seez, with the stress on the first syllable. This pronunciation is sanctioned by the Oxford English Dictionary and is used almost universally in Ireland when one is referring to the book. In his design for the cover of the 1949 Random House edition of Ulysses, the American artist Edward McKnight Kauffer emphasized the initial UL, "giving graphic form to the phonetic structure of the title with its accent on the first syllable."

- Oxford English Dictionary
- Received Pronunciation
- English language
- Greek language
- Old English
- Standard English
- New Oxford American Dictionary
- Concise Oxford English Dictionary

We see in this example that Lewis is returning results relevant to both the user selection and the context again. *Received Pronunciation*, which is regarded as the standard accent of Standard English in the United Kingdom, is directly relevant to the topic as well as the user selection. *Greek language* is also relevant to the context, as the passage is talking about pronunciation of a greek word *Ulysses*. Another interesting result is *Old English*, as this article is talking about historical pronunciation (from 1895) of the word in English.

5. RELATED WORK

Contextual exploration [13, 23] is closely related to several information retrieval problems: entity linking and search, relevance feedback, and content recommendation.

5.1 Entity Linking, Search, and Ranking

The goal of entity linking is disambiguating the mention of an entity in unstructured text to the entity's record in knowledge base, usually by using machine learning techniques and disambiguated training data. There is a rich literature, including [9, 17, 20, 27, 42]. Recently entity linking was applied to text streaming data [40] and broadcasting. [29] Contextual exploration in contrast is not limited to disambiguating an entity mention, but it also explores and recommends articles relevant to the mention as well as the context.

Entity search [31, 5] and related entity search [3] are also related to our work. Entity search aims to answer a query with direct entity answers extracted from documents or entity records in a knowledge base; related entity search/finding takes as input a query entity, the type of the target entity, and the nature of their relation, and outputs a ranked list of related entities. However, the task at hand is principally different, which is to recommend related entities that are related to the user selection, which can be any continuous

text (including but not limited to entity mentions), in the context of the document being consumed.

Entity ranking and recommendation is another recent area of research related to our work. Lee et al. [22] utilized random walk based approach for entity ranking, and Agarwal et al. [2] approached the ranking problem for networked entities with Markov walks. Vercoustre et al. [38] proposed a method for ranking Wikipedia entities.

5.2 Relevance Feedback

Relevance feedback has been shown to be effective to expand a query to relax the syntactic mis-matching problem in information retrieval [35, 34, 6]. Specifically, when a user submits a query, an IR system would first return an initial set of result documents, then ask the user to judge whether some documents are relevant or not; after that, the system would expand the query by adding a set of related terms extracted from the user's judgments, and return a set of new results. When there are no real relevance judgments available, alternatively, pseudo-relevance feedback [7, 21, 25, 14] may be performed, which simply assumes that a small number of top-ranked documents in the initial results are relevant. Our work may also be regarded as a pseudo-relevance feedback approach which assumes not only the top-ranked documents but the context around the user selection as pseudo-relevant resources. In contrast to the works above, however, we do not use syntactic query expansion. Instead, we explore a novel way of leveraging Wikipedia semantics to rank the results directly.

In the Experiments section, we showed that our approach outperforms a method based on pseudo-relevance feedback. Furthermore, since the approaches are complementary, a promising direction of future work involves the combination of IR and semantic approaches.

5.3 Content Recommendation

Content recommendation has been studied extensively in the past. Traditional content-based recommendation [33, 1] is usually to recommend documents which reflect users' long-term interests (e.g., a user might generally like sports). However, our work is recommending content related to users' ad hoc interests implied by the user selection when reading a document.

Related content recommendation [24], cumulative citation recommendation [4, 44], and contextual advertising [32] are also in the direction of ad hoc content-based recommendation. They recommend related content, such as news articles, Wikipedia articles, Web documents, or ads, to a target document. However, these works are based on a problem formulation that is insufficient for contextual exploration, as it does not allow for a user selection as part of its input.

5.4 Exploiting Knowledge Graphs

There are also several approaches that exploit the Wikipedia link-structure (or other knowledge base as a graph structure) to compute semantic similarity between text and knowledge base entities. One of the most popular areas exploiting knowledge graph structure is estimating semantic document or word similarity. WikiWalk [41] applied personalized page rank on a graph derived from Wikipedia to estimate semantic relatedness between documents. Gouws et al. [15] proposes a method for computing semantic relatedness by spreading activation energy over the hyperlink structure of Wikipedia. Mihalcea et al. [26] also presents a method for measuring the semantic similarity of short texts, using corpus-based and knowledge-based measures of similarity. Wikipedia link-structure is also exploited for computing concept relatedness [36] and query expansion [10] as well.

We emphasize that contextual entity exploration problem is a fundamentally different task from the vast bodies of work above, in that contextual entity exploration takes as input both an entity and its context. The work on exploiting semantic similarity between documents takes a document as input (our "context") and finds similar documents. However, it does not take a "pivot" entity as input (In our running example, the input entity used as pivot is "Silas Deane"). WikiWalk [41], for example, is very related to part of our approach in the sense that it performs a random walk on the Wikipedia graph. In contrast to our work, however, WikiWalk does not make a difference between the input entity and the context. Most importantly, we show in Section 4 that a baseline that consists exclusively of a random walk (like in WikiWalk) is clearly outperformed by our techniques which combine random walks with methods such as context-selection betweenness that treat the concepts of input entity and context entities as first class citizens. No problems listed above takes both an entity (user selection) and its context as input to recommend relevant entities from knowledge base.

6. CONCLUSIONS

In this paper, we presented Lewis, a system that provides a solution for the contextual entity exploration problem leveraging knowledge bases. A large scale evaluation of the approach shows significant performance improvement with respect to state-of-the art methods for contextual entity exploration. Furthermore, the results indicate that combining IR-based and knowledge-graph based methods for this problem is a promising direction of future work.

7. ACKNOWLEDGEMENTS

We thank Ashok Chandra and Panayiotis Tsaparas for their insightful feedback.

8. REFERENCES

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

[2] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[3] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the trec 2009 entity track. In *Proc. of the Text Retrieval Conference Working Notes*, 2009.

[4] K. Balog and H. Ramampiaro. Cumulative citation recommendation: Classification vs. ranking. In *Proc. of the International ACM SIGIR Conference*, 2013.

[5] I. Bordino, Y. Mejova, and M. Lalmas. Penguins in sweaters, or serendipitous entity search on user-generated content. In *Proc. of the ACM International Conference on Information Knowledge Management*, 2013.

[6] C. Buckley and S. E. Robertson. Relevance feedback track overview: Trec 2008. In *Proc. of the Text Retrieval Conference*, 2008.

[7] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using smart: Trec 3. In *Proc. of the Text Retrieval Conference*, 1994.

[8] W. Chen, W. Hsu, and M. L. Lee. Tagcloud-based explanation with feedback for recommender systems. In *Proc. of the International ACM SIGIR Conference*, 2013.

[9] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, 2007.

[10] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proc. of the International ACM SIGIR conference on Research and Development in Information Retrieval*, 2014.

[11] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *Proc. of the International World Wide Web Conference*, 2001.

[12] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[13] A. Fuxman, P. Pantel, Y. Lv, A. Chandra, P. Chilakamarri, M. Gamon, D. Hamilton, B. Kohlmeier, D. Narayanan, E. Papalexakis, and B. Zhao. Contextual insights. In *Proc. of the Companion Publication of the International Conference on World Wide Web Companion*, 2014.

[14] S. Gottipati and J. Jiang. Linking entities to a knowledge base with query expansion. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2011.

[15] S. Gouws, G. Van Rooyen, and H. A. Engelbrecht. Measuring conceptual similarity by spreading activation over wikipedia's hyperlink structure. In *Proc. of Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, 2010.

[16] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proc. of the ACM Conference on Computer Supported Cooperative Work*, 2000.

[17] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proc. of the ACM International Conference on Information and Knowledge Management*, 2012.

[18] G. Jeh and J. Widom. Scaling personalized web search. In *Proc. of the International Conference on World Wide Web*, 2003.

[19] R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar. Searching with context. In *Proc. of the International World Wide Web Conference*, 2006.

[20] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[21] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of the International ACM SIGIR Conference*, 2001.

[22] S. Lee, S.-i. Song, M. Kahng, D. Lee, and S.-g. Lee. Random walk based entity ranking on graph for multidimensional recommendation. In *Proc. of the ACM Conference on Recommender Systems*, 2011.

[23] Y. Lv and A. Fuxman. In situ insights. In *Proc. of the International ACM SIGIR Conference*, 2015.

[24] Y. Lv, T. Moon, P. Kolari, Z. Zheng, X. Wang, and Y. Chang. Learning to model relatedness for news recommendation. In *Proc. of the International World Wide Web Conference*, 2011.

[25] Y. Lv and C. Zhai. Positional relevance model for pseudo-relevance feedback. In *Proc. of the International ACM SIGIR Conference*, 2010.

[26] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proc. of the National Conference on Artificial Intelligence*, 2006.

[27] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proc. of the ACM Conference on Information and Knowledge Management*, 2007.

[28] D. Milne and I. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proc. of AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.

[29] D. Odijk, E. Meij, and M. de Rijke. Feeding the second screen: Semantic linking based on subtitles. In *Proc. of the Conference on Open Research Areas in Information Retrieval*, 2013.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[31] D. Petkova and W. B. Croft. Proximity-based document representation for named entity retrieval. In *Proc. of the ACM Conference on Information and Knowledge Management*, 2007.

[32] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. Silva de Moura. Impedance coupling in content-targeted advertising. In *Proc. of the International ACM SIGIR Conference*, 2005.

[33] S. Robertson and I. Soboroff. The trec 2002 filtering track report. In *Proc. of the Text Retrieval Conference*, 2002.

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

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

[36] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proc. of the AAAI Conference on Artificial Intelligence*.

[37] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Providing justifications in recommender systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 38(6):1262–1272, 2008.

[38] A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in wikipedia. In *Proc. of the ACM Symposium on Applied Computing*, 2008.

[39] J. Vig, S. Sen, and J. Riedl. Tagsplanations: explaining recommendations using tags. In *Proc. of the International Conference on Intelligent User Interfaces*, 2009.

[40] N. Voskarides, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Query-dependent contextualization of streaming data. In *Proc. of the European Conference on Information Retrieval*, 2014.

[41] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. Wikiwalk: Random walks on wikipedia for semantic relatedness. In *Proc. of the Workshop on Graph-based Methods for Natural Language Processing*, 2009.

[42] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum. Aida: An online tool for accurate disambiguation of named entities in text and tables. *Proc. of the VLDB Endowment*, 4(12):1450–1453, 2011.

[43] C. Yu, L. V. Lakshmanan, and S. Amer-Yahia. Recommendation diversification using explanations. In *Proc. of the IEEE International Conference on Data Engineering*, 2009.

[44] M. Zhou and K. C.-C. Chang. Entity-centric document filtering: boosting feature mapping through meta-features. In *Proc. of the ACM International Conference on Information and Knowledge Management*, 2013.