Constructing Maintainable Semantic Relation Network from Ambiguous Concepts in Web Content

Kenneth Wai-Ting Leung, Di Jiang, Dik Lun Lee and Wilfred Ng
Hong Kong University of Science and Technology

Semantic network is a form of knowledge that represents various relationships between concepts with ambiguity. The knowledge can be employed to identify semantically related objects. It helps, for example, a recommender system to generate effective recommendations to the users. We propose to study a new semantic network, namely *Concept Relation Network (CRN)*, which is efficiently constructed and maintained using existing web search engines. CRN tackles the uncertainty and dynamics of web content, and thus is optimized for many important web applications, such as social networks and search engines. It is a large semantic network for the collection, analysis and interpretation of web content, and serves as a cornerstone for applications such as web search engines, recommendation systems, and social networks that can benefit from a large scale knowledge base. In this paper, we present two applications for CRN: 1) search engine and web analytic and 2) semantic information retrieval. Experimental results show that CRN effectively enhances these applications by considering the heterogenous and polysemous nature of web content.

Categories and Subject Descriptors: H.3.0 [**Information Storage and Retrieval**]: General; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Relevance feedback*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; H.3.3 [**Information Storage and Retrieval**]: Systems and Software—*User profiles and alert services*

General Terms: Experimentation, Performance

Additional Key Words and Phrases: Concept network, semantic network, web analytic, web search, search engine, query suggestion

ACM Reference Format:
Kenneth Wai-Ting Leung, Di Jiang, Dik Lun Lee, and Wilfred Ng, 2015. Constructing Maintainable Semantic Relation Network from Ambiguous Concepts in Web Content. *ACM Trans. Internet Technol.* 9, 4, Article 39 (March 2010), 20 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Semantic network is a graphic notation for representing knowledge with interconnected nodes and arcs. Semantic networks have long been used for linguistics and machine translation purposes, which are optimized for natural language processing tasks only. They require heavy human effort to maintain the complex relationships, and thus are not effective in tackling the heterogeneous and polysemous nature of web content. To illustrate the limitation of some well-known semantic networks, we refer to Figure 1, which shows the graph representation of "apple" in Roget Thesaurus [Roget's Thesaurus 2013], WordNet [Fellbaum 1998], and YAGO [Suchanek et al. 2007]. We can see that all three semantic networks can successfully identify "apple" as a fruit. However, they cannot capture the polysemy nature of the web content which may use "apple" as a company ("Apple Computer") or the nickname of New York City ("Big Apple"), etc.

In this paper, we study a new semantic network, *Concept Relation Network (CRN)*, which tackles the uncertainty and dynamics of web content. We utilize the fact that web search engines are very effective in retrieving information on the Web. Given a query, a search engine can retrieve search

K.W.-T. Leung and D. Lee acknowledge the support of HKSAR GRF No. 615113.
Author's addresses: K.W.-T. Leung, D. Lee and W. Ng, Department of Computer Science and Engineering, Hong Kong University of Science and Technology; D. Jiang, (Current address) Baidu, Inc. China.

DOI: 0000001.0000001

results representing different semantic interpretations of the query if we examine a large enough set of results (in our experiments, we examine the top 100 results of a query). In our work, we use a concept extraction method to automatically extract concepts from the results of a query. These concepts can be considered as related concepts of the query, and by further treating each retrieved concept as a query, we can recursively retrieve related concepts of the query concepts, and thus construct a network of related concepts, called CRN, which represents different semantic interpretations of the queries.

CRN is desirable for web applications, since it can be efficiently constructed and maintained using any existing web search engine. Unlike the static approach adopted by traditional thesaurus or semantic networks [Fellbaum 1998], [Liu and Singh 2004], [Suchanek et al. 2007], by using large scale commercial search engines in the network construction, CRN is expected to have a broad coverage, and unrestricted and up-to-date vocabulary mined from a large number web pages, which are constantly crawled and updated by the search engine vendors. CRN aims to develop a large semantic network for the collection, analysis and interpretation of Web data. It can serve as a cornerstone for web applications such as web search engines [Moreno et al. 2014], recommendation systems [Yao et al. 2014] and social networks [Vosecky et al. 2014] that can benefit from a large scale knowledge base.

(a) Roget (b) WordNet (c) YAGO

Fig. 1. Knowledge representation of "apple" on Roget Thesaurus, WordNet, and YAGO

One technical issue arising from CRN is to tackle the ambiguity of concepts. A concept is ambiguous when it has several different meanings and we are uncertain about which meaning prevails in a given context. Ambiguous concepts introduce uncertainty in applications. For example, when a query is ambiguous, search engines are uncertain about what the user actually means and thus cannot produce the correct ranking for the user. CRN aims to identify possible interpretations of a concept and measure the degree of uncertainty of a concept. We adopt entropy from information theory to measure the uncertainty of a concept, which is defined based on the number for possible interpretations (i.e., related concepts) of the concept discovered by CRN. An important contribution of our work is that concepts are connected to related concepts in CRN. Thus, the uncertainty of a concept depends on not only the number of concepts it is connected to but also on the uncertainty of the connected concepts. To reflect this recursive nature of uncertainty, we introduce the *EntropySmooth* link analysis algorithm to measure the uncertainty of the concepts in CRN. Based on the premise that a concept is ambiguous if it is associated with many other ambiguous concepts, EntropySmooth iteratively smooths the entropies of the concepts in a way similar to PageRank computation.

In this paper, we propose two applications for CRN: 1) search engine and web analytic and 2) semantic information retrieval. CRN is created based on the top search results returned from the search engine. By analyzing the CRNs constructed for different search engines, we can effectively study the characteristics of different search engines. Extensive experimental evaluation is performed on the three CRNs constructed using the major commercial search engines, Google, Yahoo and Bing. We conduct extensive experiments with various measures and parameters to gauge the differences of the search engine in a principled manner. An interesting observation from our empirical result is that while Google, Yahoo and Bing are common commercial search engines which seemingly

do not differ much from each other, their constructed CRNs actually exhibit very different characteristics. For example, a major finding is that Google retrieves more focused and coherence top search results compared to Yahoo and Bing. On the other hand, Yahoo and Bing's top search results provide broader and more diversified information coverage.

CRN can be used to infer the semantic relevance of a document based on the background knowledge stored in CRN. Traditional search engines rely very much on exact or partial matching between keywords but not their semantics. Further, since search queries are usually short and ambiguous, CRN can be used to effectively determine concepts that are semantically related to the user query based on concepts that form strongly connected subgraphs of CRN. We propose algorithms to determine semantically related concepts from CRN to illustrate the practicality of our approach. Experimental results show that our proposed random walk algorithm can effectively retrieve semantically terms from CRN for semantic information retrieval.

To the best of our knowledge, our work is the first that employs search engines to construct a semantic network. The main contributions of this paper can be summarized as follows:

— Our proposed *Concept Relation Network (CRN)* is automatically constructed and self-maintainable using existing web search engines. We show that semantically related concepts can be easily identified by studying the connections of the concepts in CRN.
— CRN can be used to effectively discover concepts and their relations from people's writings, and thus it can serve as a knowledge base with broad applications.
— We apply the notion of entropy to measure the ambiguity of concepts in CRN and develop an iterative algorithm, *EntropySmooth*, to smooth the entropy values. Experiments show that the smoothed entropy values are effective in determining the randomness of the search results being retrieved by a query.
— We propose to apply CRN to search engine and web analytic, and perform extensive experimental evaluation on the three CRNs constructed using Google, Yahoo and Bing. This gains interesting insights of major commercial web search engines.
— We also propose to apply CRN to semantic information retrieval, and propose a random walk algorithm to effectively determine semantically related concepts from CRN.

The rest of the paper is structured as follows. Related work is reviewed in Section 2. Section 3 presents the preliminaries of extracting important concepts on the web using search engines. In Section 4, we introduce an iterative algorithm to construct Concept Relation Network (CRN) using search engines. In Section 5, we propose several parameters to evaluate three CRNs constructed using the three major commercial search engines. In Section 6, we present our strategy of discovering semantically related concepts. Section 7 gives our evaluation results. Finally, Section 8 concludes the paper.

Table I. Top Concepts for "apple" extracted from different search engines

Google	Yahoo	Bing
apple store	apple computer	apple consultant network
apple computer	app store	apple retail store
iphone	leave of absence	ipad
apple product	health	ferry halim
mac	news of steve job	datum recovery

2. RELATED WORK

In this section, we first review some semantic networks that are commonly used in web applications. Then, we review existing work related to search engine and web analytic. Finally, we discuss existing work related to the application of semantic networks on discovering semantically related terms for semantic information retrieval.

2.1. Semantic Networks

One well-known semantic network is WordNet [Fellbaum 1998]. The main goal of WordNet is to support intelligence systems that involve automatic text analysis. It is a lexical database which groups English words into sets of synonyms called *synsets*. It provides a short description on each synset, and maintains different semantic relationships among the synsets. Another well-known semantic network is ConceptNet [Liu and Singh 2004], which organizes words into a relational ontology. ConceptNet is a commonsense knowledgebase and natural-language-processing corpus that supports textual-reasoning tasks. More recently, YAGO [Suchanek et al. 2007] was developed to extract facts from Wikipedia and WordNet based on a combination of rule-based and heuristic methods. In [Yazdani and Popescu-Belis 2013], a method was proposed to compute similarity between words and texts using knowledge in a hypertext encyclopedias, such as Wikipedia. A network of concept is constructed by filtering encyclopedia's articles, and each concept corresponds to an article in the encyclopedia. We observe that most existing semantic networks (e.g., WordNet, ConceptNet, and YAGO) contain rich lexical knowledge. However, they require heavy human effort or an external knowledge base, such as Wikipedia, to maintain the complex relationships. In contrast, our proposed Concept Relation Network (CRN) can be automatically constructed and self-maintained by using existing web search engines. Furthermore, it captures the heterogeneous and polysemous nature of web content by considering unrestricted and up-to-date vocabulary mined from a large number web pages, which are constantly crawled and updated by the search engine vendors.

2.2. Search Engine and Web Analytic

For search engine analysis and evaluation, Lawrence, et. al. performed a study of the coverage of six major search engines with respect to the total number of documents on the web in [Lawrence and Giles 1998]. Later on, Lawrence, et. al. [Lawrence and Giles 2000] repeated the same experiments on 11 major search engine and found that the estimated size of the web increased significantly from 320 million to 800 million pages. More recently, Chowdhury, et. al. [Chowdhury and Soboroff 2002] compared the retrieval effectiveness of different search engines and discovered that some search engines can retrieve more relevant results compared to the others for some specific queries. We observe that most of the existing work focuses on statistical analysis (e.g., no. of relevant documents) of the information retrieved by different search engines, while lacking content analysis (e.g., the diversity of the search results, etc). Thus, we propose to construct CRNs using commonly used commercial search engines and perform a large-scale content analysis on the constructed CRNs.

2.3. Semantic Information Retrieval

Many recent studies utilize search context in the query suggestion process. In [Liao et al. 2011], Liao, et. al. also proposed a novel context-aware query suggestion approach by mining concept sequences from search logs. Leung, et. al. [Leung et al. 2008] proposed online techniques to extract concepts from the search results, and the extracted concepts are used to identify suggestions for the target query. Jain, et. al. [Jain et al. 2011] proposed a novel way of generating query suggestions by moving beyond the dependency on search query logs and providing synthetic suggestions for web search queries. The synthetic suggestions are generated upon novel query-level operations and combines information available from various textual sources. More recently, Gao, et. al. proposed a new query expansion method based on path-constrained random walks in [Gao et al. 2013]. These approaches resolve the following problem: if the query suggestions are generated based on query logs, the quality of suggestions becomes too much dependent on the quality of the queries submitted by the users. Thus, if there are many novice users, the queries in the query logs are of low quality, making it hard to obtain high quality suggestions from the logs. Our CRN also generates query suggestions without using query logs and enjoys similar benefits of the above work.

3. PRELIMINARIES

In this section, we introduce our concept extraction method that discovers important topics related with an input search query. We define two kinds of entropies, namely content and location entropies, to measure the ambiguity of the content and location information retrieved by using the input query.

3.1. Concept Extraction

We assume that if a keyword/phrase exists frequently in the web-snippets[1] arising from the query q, it represents an important concept related to the query, as it co-exists in close proximity with the query in the top documents. Thus, our content concept extraction method first extracts all the keywords and phrases from the web-snippets arising from q. The extracted keywords and phrases are referred to as the set of candidate concepts for q. After obtaining a set of candidate concepts, we employ the following support formula, which is inspired by the problem of finding frequent item sets in data mining [Church et al. 1991], to measure the interestingness of a concept c_i with respect to q:

$$threshold < support(c_i) = \frac{sf(c_i)}{n} \cdot |c_i| \tag{1}$$

where sf is the snippet frequency of a concept, n is the total number of snippets and $|c_i|$ is the length of the concept c_i. The threshold is set to 0.03 after experimentation in our preliminary study [Leung et al. 2011b]. A small threshold is chosen in order to include as many concepts as possible into CRN, while eliminating concepts that are too rare to be significant.

Concepts can be further categorized (e.g., location names, people's names, financial terms, etc). However, we only utilize *content concepts* and *location concepts* in CRN, since they provide important topical and locational information associated with the extracted concepts. E.g., if the concept "apple" is associated with another concept "iphone", then we know that "apple" is related to a product from Apple Computer, and if the concept "apple" is closely related to a location "New York", then we know that "apple" may be referred as the nickname, "Big Apple", of New York. In general, concepts extracted from web-snippets are referred to as *content concepts*. Obviously, content concepts extracted from different search engines are different even for the same query, since different search engines return different top search results. Thus, by comparing and analyzing the concepts extracted from the top results, we can effectively study the "searching behavior" of different web search engines. Table I shows the example concepts extracted for the query "apple" using Google, Yahoo, and Bing (as of 26-01-2011)[2]. From the extracted concepts, we can see that Google retrieved a set of more focus top results about "Apple Computer" for the query apple. On the other hand, Yahoo retrieved top results related to recent news about the CEO of "Apple Computer". Finally, Bing retrieved a wider range of results related to many different topics, such as "Apple Computer", games, and recovery software. By representing the unstructured search results as CRNs, comparison of search engines in terms of retrieval effectiveness and information coverage becomes feasible, even on the huge volume of web data.

We now discuss another important type of concepts. *Location concepts* are location information associated with the input query. A concept is considered as a location concept, if it matches a geographic name contained in National Geospatial [Geographic Names for Geopolitical Areas from GNS 2013] and World Gazetteer [World Gazetteer 2013] covering countries and geopolitical areas. Location concepts are relatively more stable and are very often attached to search queries to confine the location scope of the result. In our experiments, the extracted concepts matched $17,000$ city, province, region, and country names from National Geospatial and World Gazetteer. Each match will make the concept a location concept for the input query. The relationships between different

[1] "Web-snippet" denotes the title, summary and URL of a Web page returned by search engines.

[2] Although Bing started serving Yahoo starting from SEP 2010 (http://betanews.com/2010/09/14/bing-overtakes-yahoo-and-that-s-not-a-good-thing/), we observe that their search results are still different due to the variance of ranking functions

locations are also provided by National Geospatial and World Gazetteer. Specifically, all cities are organized as children under their provinces, all the provinces are organized as children under their regions, and all the regions are organized as children under their countries.

3.1.1. Content Entropy. In information theory, the notion of entropy formalizes the uncertainty associated with the information content of a message from the receiver's view point. In the context of web search, entropy can be employed to denote the uncertainty associated with the information content of the search results from the user's point of view. The higher the entropy of a query, the higher the topical randomness of the search results being retrieved by the query. Some queries may induce the extraction of a larger number of content concepts. For example, queries such as "mp3" may retrieve search results ranging from "blog", "band", "software", "download" to "ipod". This shows that "mp3" is an ambiguous query, since it is associated with many different concepts. On the other hand, the query "uno" returns search results about a card game, and thus there is little diversity observed on the content concepts extracted from the search results. In our experiment, only 18 content concepts were extracted for the query "uno", but 49 content concepts were extracted for the query "mp3".

We emphasize that, there is no preference of a high entropy value to a low one, or vice versa. It is simply a measure of the diversity of information content. The following formula is used to compute the content entropies $H_C(q)$ of the concept retrieved for q.

$$H_C(q) = -\sum_{i=1}^{k} p(c_i) \log_2 p(c_i) \tag{2}$$

where k is the number of extracted content concepts $C = \{c_1, c_2, ..., c_k\}$, $|c_i|$ is the number of search results containing the content concept c_i, $|C| = |c_1| + |c_2| + ... + |c_k|$ and $p(c_i) = \frac{|c_i|}{|C|}$.

$H_C(q)$ can be quite different across different search engines, because they retrieve different sets of cocnepts. The different results give rise to different CRNs for different search engines in our analysis.

3.1.2. Location Entropy. We now compute the location entropy of an input query. The computation is similar to that of the content entropy, except that only location concepts are considered.

$$H_L(q) = -\sum_{i=1}^{m} p(l_i) \log p(l_i) \tag{3}$$

where m is the number of extracted location concepts $L = \{l_1, l_2, ..., l_m\}$, $|l_i|$ is the number of search results containing the location concept l_i, $|L| = |l_1| + |l_2| + \cdots + |l_m|$, and $p(l_i) = \frac{|l_i|}{|L|}$.

Comparing with content entropies, location entropies obtained from the sample queries have a wider range of variations. As shown in in Table **??**, some queries receive location entropies as high as 10.08 and as low as 6.47. For example, the query "apartment" receives a high $H_L(q)$ of 9.50, since the query returns pages that are associated with many different locations, specifically a total of 58 locations concepts. However, some queries such as "window vista" and "shareware" retrieve mainly different content information from tips and skills to online download and support. These queries are associated with only a few locations and thus, they receive relatively low $H_L(q)$.

4. CONCEPT RELATION NETWORK

In this section, we first discuss the construction of CRN. Then, we introduce a link analysis algorithm EntropySmooth that smooths the content and location entropies of the concepts.

4.1. Concept Relation Network Construction

To construct CRN, we start with a query[3] to extract concepts from the the top web-snippets. A concept extracted from the query can also serve as a query to retrieve another set of concepts related to the query. Detailed of the CRN construction algorithm can be found in our preliminary study [Leung et al. 2011a]. The input to the CRN construction algorithm is a queue containing the initial queries. In our experiments, we employ a set of 250 initial queries from different topics to make sure that the constructed CRN can cover topics of different categories on the web. During the CRN construction, it is to retrieve a concept that has already been retrieved in a previous iteration or by a previous query. Thus, a concept may point back to any of the existing concepts in the graph. In contrast to the construction of hierarchical user profiles in [Xu et al. 2007], we now obtain a huge interconnected graph instead of a small tree. Figure 3 shows a small fraction of the CRN graph around "ontology". The links between concepts are not fully shown here for clarity. However, we can still observe that there are diversified ways to connect a concept with other nodes within level 2. In general, level i has more concepts than level $(i-1)$.

$$PC(q, c_i) = p(c_i|q) = \frac{support(c_i)}{support(q)}. \tag{4}$$

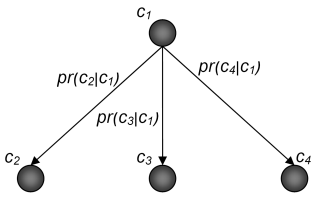


Fig. 2. A sample branch showing c_2, c_3 and c_4 as the child nodes of c_1.

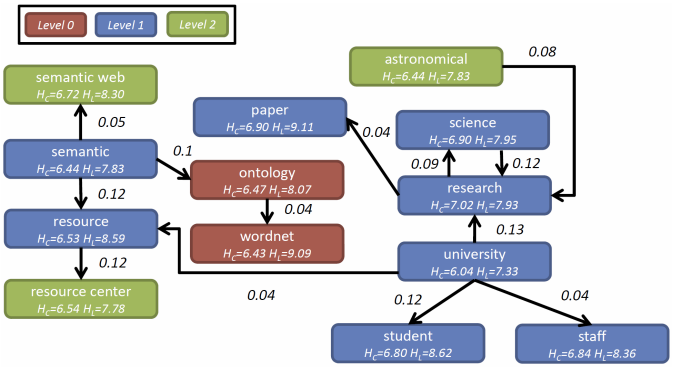


Fig. 3. A fraction of the CRN graph built around the concept "ontology"

4.2. Entropy Smoothing

The initial entropies obtained from Equations (2) and (3) only take into account the number of retrieved concepts and their occurrence probabilities. However, a concept may be *ambiguous* and this factor should be considered in building CRN. We propose that if a concept retrieves *ambiguous* concepts, the concept itself should also be ambiguous. We call this the *ambiguity proposition*. In other words, the ambiguity of a concept depends not only on the number of retrieved concepts, as implied by the original entropy formulas, but also on the ambiguity of the retrieved concepts. We now develop an iterative algorithm, EntropySmooth, which shares the spirit of PageRank [Page et al. 1999], to smooth the initial entropy values.

The PageRank algorithm assumes that a node will get a high PageRank, if it has pages with high PageRank pointing to it. The assumption is similar to our *ambiguity Proposition*. Comparatively, our model focuses on outgoing links (an ambiguous concept can retrieve many other ambiguous concepts), while the PageRank model focuses on incoming links (an authority page is pointed at by many other authority pages).

[3]Since a query is also regarded as a concept in CRN, we sometimes refer to a query as a *query concept* or simply a *concept* when no ambiguity arises.

Figure 2 shows an example of CRN that has the concept c_1 being the parent of the concepts c_2, c_3, and c_4. According to the ambiguity proposition, c_1 should have high entropy value if its retrieved concepts (i.e. c_2, c_3, and c_4) also have high entropy values. We now let the initial entropies of c_1, c_2, c_3, and c_4 computed by using Equation (2) be $H_C(c_1)$, $H_C(c_2)$, $H_C(c_3)$, and $H_C(c_4)$. In order to propagate the entropies of c_2, c_3, and c_4 to that of c_1, we compute the *EntropyScore* of c_1, denoted as $CS(c_1)$, as follows.

$$\begin{aligned} CS(c_1) = (1-d_C)H_C(c_1) + d_C(H_C(c_2)\cdot pr(c_2|c_1) \\ +H_C(c_3)\cdot pr(c_3|c_1) + H_C(c_4)\cdot pr(c_4|c_1)) \end{aligned} \tag{5}$$

where $pr(c_2|c_1)$, $pr(c_3|c_1)$, and $pr(c_4|c_1)$ are the probabilities of c_2, c_3, and c_4 existing as concepts in the search results of c_1.

For example, let both c_2 and c_3 appear once while c_4 appears twice in the search results of c_1. Then $pr(c_2|c_1) = \frac{1}{1+1+2} = 0.25$, $pr(c_3|c_1) = \frac{1}{1+1+2} = 0.25$, and $pr(c_4|c_1) = \frac{2}{1+1+2} = 0.5$, which sum up to 1 (i.e., $(pr(c_2|c_1) + pr(c_3|c_1) + pr(c_4|c_1)) = 1$). $H_C(c_i))$ is the initial entropy of c_i computed by using Equation (2). d_C is a damping factor that balances the contributions between the initial entropy $H_C(c_1)$ of c_1 and the initial entropies, $H_C(c_2)$, $H_C(c_3)$ and $H_C(c_4)$, which are propagated from c_2, c_3 and c_4.

A major impact of introducing EntropySmooth on concept extraction is that the entropies no longer rely solely on the extracted concepts at search time. Thus, even if a concept gains too high or low initial content entropy or location entropy due to some noisy concepts extracted from the search results, the entropies can still converge to a stable point.

4.2.1. Content EntropyScore $CS(c)$. Assume that there are n concepts in the CRN, with H_C as a vector containing the initial content entropies (as already discussed in Section 3.1.1) of the concepts as follows.

$$H_C = \begin{pmatrix} H_C(c_1) \\ H_C(c_2) \\ H_C(c_3) \\ ... \\ H_C(c_n) \end{pmatrix}$$

Let A be a matrix containing the relationships between two concepts c_i and c_j as $pr(c_j|c_i)$ that if there is an edge from c_i to c_j in the CRN. In other words, we fill the item in the i^{th} row and j^{th} column with $pr(c_j|c_i)$ in A as follows.

$$A = \begin{pmatrix} 0 & pr(c_2|c_1) & pr(c_3|c_1) & ... & pr(c_n|c_1) \\ pr(c_1|c_2) & 0 & pr(c_3|c_2) & ... & pr(c_n|c_2) \\ pr(c_1|c_3) & pr(c_2|c_3) & 0 & ... & pr(c_n|c_3) \\ ... & ... & ... & ... & ... \\ pr(c_1|c_n) & pr(c_2|c_n) & pr(c_3|c_n) & ... & 0 \end{pmatrix}$$

where the conditional probabilities of each row sum up to 1 (i.e., $\sum_{j=1}^{n} pr(cj|c_i) = 1$).

Given the above initial entropy vector H_C and the relationship matrix A, and a damping factor d_C, the content EntropyScore CS vector of the concepts is iteratively updated as follows.

$$CS_{i+1} = (1-d_C)H_C + d_C(A\cdot CS_i) \tag{6}$$

where the initial entropies in H_C are used as the initial content EntropyScore CS_0 ($CS_0 = H_C$). The resulting content Entropy Scores in Equation (6) quantify the ambiguity of the concepts in the

CRN. If a concept c_i is ambiguous, which intuitively means that c_i has many meanings, then $CS(c_i)$ should be high, and vice versa.

4.2.2. Location EntropyScore $LS(c)$. As discussed in Section 3.1.2, a concept can also be associated with an initial location entropy $H_L(c_i)$, representing the diversity of location information associated with the search results. The *ambiguity proposition is also applicable to location concepts: if a concept is location ambiguous, it retrieves many concepts with high location entropies.* Thus, given the initial location entropy vector H_L, the relationship matrix A, and a location damping factor d_L, we can also compute the location EntropyScore $LS(c)$ using the content EntropyScore method proposed in the previous section.

5. APPLICATION I: SEARCH ENGINE AND WEB ANALYTIC

In this Section, we propose various CRN measures and parameters, which can be employed to gauge the quality and quantity of the information coverage of a search engine.

5.1. Degree of Information Coverage

To measure the amount of retrieved information with respect to a given query, we have to analyze the amount of information being retrieved and the ambiguity of the information being retrieved. Table II shows lists of concepts extracted for the query "apple". We denote by $CL1$, $CL2$, and $CL3$ the three different lists of concepts. Obviously, if a search engine SE retrieves many different concepts for a particular query (e.g., "apple"), it has a good information coverage on the query "apple". From this perspective, $CL3$ thus has a better information coverage comparing to $CL1$. In addition, we observe that the ambiguities of the concepts are also important in measuring information coverage since ambiguous concepts in turn retrieve more diversified information. For example, $CL1$ and $CL2$ both contain 5 concepts. However, $CL2$'s search results contain richer information content, since they contain several ambiguous concepts, such as "sick". From this observation, we define the quantitative measure $IC_C(q, SE)$ should take both the number of concepts being extracted from a query q and content entropy into consideration. We take the sum of the content entropies of the retrieved concepts as a measure of the *content information coverage* $IC_C(q, SE)$ (i.e., the diversity of the information retrieved), and normalize it against the maximum IC_C among all the queries to make sure that $IC_C(q, SE)$ lies within $[0, 1]$.

$$IC_C(q, SE) = \frac{\sum_{i=1}^{n} CS_{SE}(c_i)}{MAX\_IC_C(SE)} \tag{7}$$

where $c_1, c_2, c_3, \ldots, c_n$ are the concepts retrieved for the query q submitted to the search engine SE. $MAX\_IC_C(SE)$ is the maximum IC_C that is obtained from SE and is utilized here for normalization. Equation 7 measures the quantity of *content* information coverage of SE on q, since only *Content EntropyScore* $CS_{SE}(c)$ is used. Similarly, we measure the quantity of *location information coverage* $IC_L(q, SE)$ of SE on q using *Location EntropyScore* $LS_{SE}(c)$ as follows:

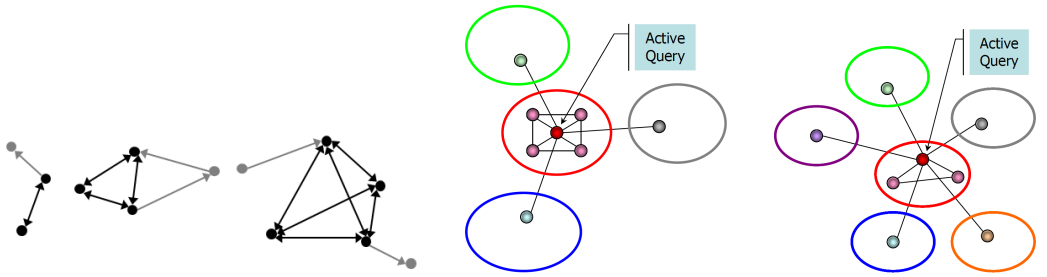
$$IC_L(q, SE) = \frac{\sum_{j=1}^{n} LS_{SE}(c_i)}{MAX\_IC_L(SE)}. \tag{8}$$

5.2. Degree of Coherence

Semantically related concepts can be straightforwardly discovered by performing link analysis on CRN. More specifically, concepts that are semantically related to one another can be discovered by finding *complete subgraphs* in CRN. With directed links, the requirement is stricter, where each pair of nodes should point to each other. Figure 4(a) illustrates complete subgraphs with sizes 2, 3, and 4. A complete–link single–pass clustering algorithm is used to obtain a set of complete subgraphs. The similarity between two concepts is defined by the sum of the parent-child scores, as given by Equation 4, between the two concepts in both directions.

Table II. Example of calculating the information coverage arising from the query “apple”

CL_1	CS_1	LS_1	CL_2	CS_2	LS_2	CL_3	CS_3	LS_3
computer	6.682	8.944	computer	6.682	8.944	computer	6.682	8.944
apple store	6.680	8.762	apple store	6.680	8.762	apple store	6.680	8.762
iphone	6.560	8.601	steve job	6.255	8.736	iphone	6.560	8.601
ipad	6.612	8.513	sick	7.199	9.021	ipad	6.612	8.513
mac	6.640	7.886	mac	6.640	7.886	mac	6.640	7.886
						steve job	6.255	8.736
						sick	7.199	9.021
$IC_C(CL_1)$	0.033		$IC_C(CL_2)$	0.033		$IC_C(CL_3)$	0.046	
$IC_L(CL_1)$		0.043	$IC_L(CL_2)$		0.043	$IC_L(CL_3)$		0.061



(a) An illustration of complete subgraphs with different sizes

(b) High quality cluster which is coherent

(c) Low quality cluster which is loosely coupled

Fig. 4. Examples of complete subgraphs, high quality and low quality clusters

To measure the degree of coherence of the information retrieved by a query q using a search engine SE, we analyze the complete subgraphs (i.e., the semantic clusters) related to q using the CRN for SE. If SE is efficient in retrieving information related to q, it should be able to retrieve information that is strongly related to q (i.e., the intra-similarity within the semantic cluster for q should be high). On the other hand, it should also be able to distinguish ambiguous concepts that are shared by other semantic clusters (i.e., the inter-similarity between different semantic clusters should be low).

Figure 4 illustrates examples of a highly coherent cluster and a loosely coupled cluster. Figure 4(b) shows an example of a highly coherent cluster, in which semantically related concepts are tightly connected to one another, while concepts that are shared by other clusters are loosely connected to the active query q. In contrast, Figure 4(c) shows an example of a loosely coupled cluster, in which less semantically related concepts are connected to the active query, and many of the concepts connected to the active query q are shared by other clusters.

In order to optimize the degree of coherence of the information retrieved by q using SE, the intra-similarity of concepts c (i.e., $PC(q, c_i)$) within the semantic cluster for q should be maximized, while the inter-similarity (i.e., $PC(q, c'_i)$) between q and concepts c' in other semantic clusters should be minimized. Thus, we measure *the degree of coherence* $CO(q, SE)$ of the information by dividing the intra-similarity with the inter-similarity, and normalize it against the maximum CO among all the queries to make sure that $CO(q, SE)$ lies within $[0, 1]$ as follows.

$$CO(q, SE) = \frac{\frac{\sum_{i=1}^{n} PC(q,c_i)}{\sum_{j=1}^{m} PC(q,c'_j)}}{MAX\_CO(SE)} \tag{9}$$

where $c_1, c_2, c_3, ..., c_n$ are the concepts within the same cluster for the query q using the search engine SE, and $c'_1, c'_2, c'_3, ..., c'_m$ are the concepts connecting to q in other clusters. $MAX\_CO(SE)$ is the maximum CO that is obtained from SE and is utilized here for normalization. The above

formula measures the quality of information covered by q using SE, the higher the $CO(q, SE)$, the better the quality of information covered by q using SE.

5.3. Degree of Coverage and Coherence C^2

After the information coverage $IC_C(q, SE)$, $IC_L(q, SE)$ and the coherence measure $CO(q, SE)$ are computed for a given query q, they can be used to analyze the characteristics of the search results. For example, SE_1 may retrieve a large amount of loosely coupled for q; while SE_2 may retrieve a small amount of highly coherence search results for q. To obtain a single measure to analyze the characteristics of the search results obtained by SE for q, we propose the following *Coverage and Coherence measure* $C^2(q, SE)$ parameter to measure both the amount information coverage and the degree of coherence the search results obtained by SE for q. Basically, if SE retrieves a large amount of coherent search results for q, $C^2(q, SE)$ will be high. On the other hand, if SE retrieves a small amount of loosely coupled search results for the q, $C^2(q, SE)$ will be low. Notably, the SECA value for some queries is zero, since those queries do not belong to any complete subgraphs, resulting in $CO(q, SE) = 0$.

Since a good search engine should have good information coverage (i.e., high IC) and highly coherent retrieved results (i.e., high CO), we propose the following SECA score, which is the product of the information coverage (i.e., $IC_C(q, SE) + IC_L(q, SE)$) and the degree of coherence (i.e., $CO(q, SE)$).

$$C^2(q, SE) = CO(q, SE) \cdot (IC_C(q, SE) + IC_L(q, SE)) \tag{10}$$

where $CO(q, SE)$ is the degree of coherence of the search results retrieved by q using SE, $IC_C(q, SE)$ is the *content* information coverage of SE on q and $IC_L(q, SE)$ is the *location* information coverage of SE on q.

6. APPLICATION II: DISCOVERING SEMANTICALLY RELATED CONCEPTS FOR SEMANTIC INFORMATION RETRIEVAL

In this section, we discuss the strategy of discovering semantically related concepts. Since CRN contains concepts mined using web search engines, one straight-forward application is to identify semantically related terms from CRN for semantic information retrieval. In this paper, we propose an algorithm to efficiently determine semantically related concepts from CRN. The detailed algorithm for retrieving the concepts is presented in Algorithm 1. The underlying logic of Algorithm 1 is as follows: Given an input query, we first tokenize it into a set of terms (e.g., "$applecomputer''$, it will be tokenized into $S =$ "$apple''$, "$computer''$). By utilizing these terms as the starting points, we can easily retrieve concepts that are semantically related to the input query from CRN. Our experimental results in Section 7.6 confirm that CRN is an effective semantic network, which significantly outperforms a baseline semantic network in retrieving semantically relevant terms for web applications.

7. EXPERIMENTS

In this section, we evaluate the CRNs constructed by using Google, Yahoo, and Bing. In Section 7.1, we present the setup for constructing the CRNs. In Section 7.2, we analyze the Google CRN, Yahoo CRN, and Bing CRN using the parameters presented in Section 5. We evaluate the entropy scores obtained from CRN in Sections 7.3 and 7.4. We then evaluate *complete subgraph* and *closely linked subgraph* for determining semantically related concepts from the CRNs in Section 7.5. Finally, we evaluate the effectiveness of the proposed random walk algorithm in retrieving semantically related concepts in Section 7.6.

7.1. Experimental Setup

We implemented the CRN with Java. The data structure of CRN is a graph in general, and we implemented it through ArrayList in Java. Since we only store the nodes and edges in CRN, the CRN

ALGORITHM 1: *retrieveQueryTerms*(search query q, link threshold T, maximum steps N)

Input: q, T, N
Output: G_c
Tokenize q as $S = (t_1, t_2, ..., t_m)$;
$stepCounter = 0$;
$G_c = \{\}$;
while *stepCounter*$< N$ **do**
  forall the $t \in S$ **do**
    $C \leftarrow$ concepts reached by 1 step of walk on CRN with t as the startpoint;
    forall the $c \in C$ **do**
      if $Link(t, c) > T$ **then**
        $G_c = G_c \cup \{c\}$;
      end
    end
    $S = G_c$;
  end
  $stepCounter++$;
end
return G_c;

Table III. CRN Sizes

Search Engine	# lv0 node	# lv1 node	# lv2 node
Google	250	4575	37674
Yahoo	250	4074	31095
Bing	250	4522	27147

Table IV. CRN Links Between Nodes at Different Levels

	Google	Yahoo	Bing
# lv 0→0 links	350	188	339
# lv 0→1 links	15735	8799	16726
# lv 0→2 links	0	1	0
# lv 1→0 links	6185	3236	5864
# lv 1→1 links	263931	105470	268790
# lv 1→2 links	76502	43791	34742
# lv 2→0 links	42340	22966	35305
# lv 2→1 links	2062060	722717	1574163
# lv 2→2 links	532071	234184	172212

construction process does not require a huge amount of memory (a Pentium machine with 2 GB memory was used for the construction process). The concept extraction process is performed offline, it took approximately 11 to 12 hours for the CRN construction. In the subsequent experiments, 250 concepts are randomly selected as initial queries (i.e., the concepts in level 0) from 16 different top-level Open Directory Project categories in order to ensure that topics from different perspectives are well covered. In the concept extraction phase, the top 100 snippets returned from a query were used in the analysis. To control the growth of CRN, we terminated the concept extraction iteration at level 2. We already obtained a significantly large number of concepts. Table III summarizes the number of nodes on each level of the three CRNs.

The initial queries are selected from different categories in order to make them succinct and interesting to a large number of users. One may think that the resultant CRN structure is likely to be a tree with great breadth. However, the statistics in Table IV shows that many higher level nodes actually point back to their previous nodes, meaning that many concepts retrieved in a later iteration had already been retrieved in an earlier iteration. Thus, the CRNs are a large interconnected graph rather than a tree. Table IV summarizes the distribution of the links between the nodes at different levels. We observe that the number of nodes increases rapidly as the number of level increases, and the level-1 nodes are referred by most other concepts.

Table V. Content Information Coverage Analysis (Average $IC_C(q, SE)$)

Search Engine	Level 0	Level 1	Level 2	Overall
Google	0.49703	0.52994	0.48965	0.49398
Yahoo	0.24939	0.23275	0.19196	0.19692
Bing	0.54439	0.49357	0.42419	0.43480

Table VI. Location Information Coverage Analysis (Average $IC_L(q, SE)$)

Search Engine	Level 0	Level 1	Level 2	Overall
Google	0.61491	0.65540	0.60547	0.61084
Yahoo	0.31690	0.29586	0.24403	0.25034
Bing	0.61158	0.55450	0.47660	0.48851

7.2. Content Analysis

We now analyze Google CRN, Yahoo CRN, and Bing CRN by using the parameters $IC_C(q, SE)$, $IC_L(q, SE)$, $CO(q, SE)$, and $C^2(q, SE)$. First, we analyze the content information coverage of the three search engines using the average $IC_C(q, SE)$. We observe in Tables III and V that even Bing has the least number of concepts in its CRN, it has a broad information coverage with average $IC_C(q, Bing) = 0.43480$, since Bing's search results contain relatively diverse information. On the other hand, we observe that Google CRN yields the highest average $IC_C(q, Google) = 0.49398$ while Yahoo CRN yields the lowest average $IC_C(q, Yahoo) = 0.19692$. Similar to the results of $IC_L(q, SE)$, Google achieves the highest $IC_L(q, SE) = 0.61084$, while Yahoo has the lowest $IC_L(q, SE) = 0.25034$, as shown in Table VI.

We proceed to evaluate the degree of coherence of the retrieved concepts for the three search engines. As discussed in Section 5, we need to first discover the complete subgraphs in the CRNs before computing $CO(q, SE)$ for the three search engines. Table VII shows the statistics of different sized complete subgraphs in the three CRNs. We observe that Google has the largest number of complete subgraphs. This shows that the concepts retrieved by Google are closely related to one another, and thus yielding more complete subgraphs. After constructing all the complete subgraphs, we compute the $CO(q, SE)$ for the three search engines, and the results are shown in Table VIII. We observe that Google CRN yields a high degree of coherence ($CO(q, Google) = 0.0291$) compared to the other two CRNs ($CO(q, Yahoo) = 0.0083$ and $CO(q, Bing) = 0.0073$). This again shows that most of the concepts in Google CRN are closely connected to one another, and thus its search results are more focused and coherent than the other two. Notice that the underlying reason of Google CRN having a large number of complete subgraphs is not entirely due to its size. The size of Google CRN is only 1.2 times of Yahoo CRN and 1.33 times of Bing CRN. However, the number of complete subgraphs in Google CRN is 6.15 times of that of Yahoo CRN and 4.98 times of that of Bing CRN.

We compute $C^2(q, SE)$ for the three search engines, and present the results in Table IX. Since Google yields both high $IC_C(q, SE)$, $IC_L(q, SE)$, and $CO(q, SE)$, it achieves the highest average C^2 score compared to the other two search engines. Another observation is that all the three search engines achieve high $CO(q, SE)$ and $C^2(q, SE)$ scores at level 1, but the scores gradually decrease from level 1 to level 2, and also from level 2 to level 3. This phenomenon is caused by the strict definition of degree of coherence. To remedy the situation, we will develop a relaxed version of $CO(q, SE)$ to further investigate the relationships between concepts in CRNs in Section 7.5.

Table VII. Statistics of Complete Subgraphs

	Google	Yahoo	Bing
# Subgraphs	978	159	196
Maximum Subgraph Size	7	4	6
# Size 3 Subgraph	822	153	180
# Size 4 Subgraph	130	6	14
# Size 5 Subgraph	18	0	1
# Size 6 Subgraph	6	0	1
# Size 7 Subgraph	2	0	0
Average Subgraph Size	3.19	3.04	3.10
Average Co-occurrence Score	0.4283	0.4971	0.4380

Table VIII. Degree of Coherence Analysis (Average)

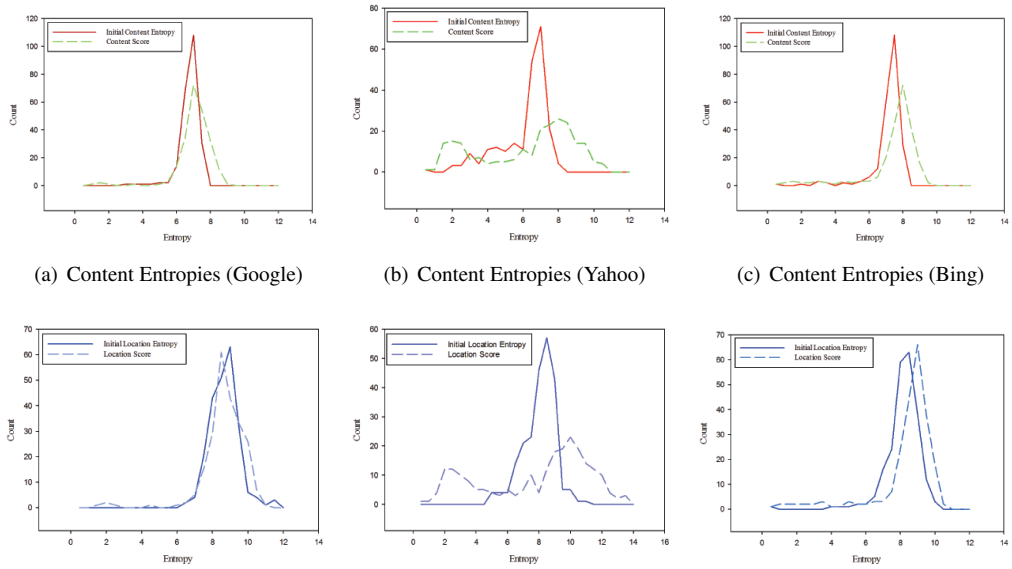
Search Engine	Level 0	Level 1	Level 2	Overall
Google	0.0928	0.0814	0.0223	0.0291
Yahoo	0.0299	0.0321	0.0049	0.0083
Bing	0.1003	0.0252	0.0036	0.0073

Table IX. C^2 Analysis (Average)

Search Engine	Level 0	Level 1	Level 2	Overall
Google	0.11352	0.09926	0.02409	0.03266
Yahoo	0.01696	0.01857	0.00243	0.00437
Bing	0.06438	0.02739	0.00362	0.00741

7.3. Entropy Analysis

Figures 5(a) to 5(c) show the *smoothed* distribution of the content entropies of the initial query concepts (i.e., the concepts at level 0) before and after running EntropySmooth. Figures 5(d) to 5(f) show the *smoothed* distribution of the location entropies of the initial query concepts (i.e., concepts at level 0) before and after running EntropySmooth. The smoothing is done by spreading the entropy value of a concept to its neighboring concepts. In Figures 5(a), 5(b) and 5(c), we observe that initial content entropies $H_C(c)$ are less stable and the content Entropy Scores $CS(c)$ computed using EntropySmooth induces a spread of the peak, with the average remains the same throughout. The same observation can also be derived from the initial location entropies $H_L(c)$ and the location Entropy Scores $LS(c)$, as shown in Figures 5(d) to 5(f).



(a) Content Entropies (Google) (b) Content Entropies (Yahoo) (c) Content Entropies (Bing)

(d) Location Entropies (Google) (e) Location Entropies (Yahoo) (f) Location Entropies (Bing)

Fig. 5. The distribution of entropies before and after EntropySmooth

Tables X shows example Content Scores $CS(c)$ and Location Scores $LS(c)$ obtained from the three CRNs. The symbol "$\uparrow$" means that the Entropy Score has increased comparing to the initial entropy, "$\simeq$" means that the Entropy Score stays roughly the same as the initial entropy, and "$\downarrow$" means that the Entropy Score has decreased comparing to the initial entropy.

The $\uparrow$ concepts, such as "coca cola bear", "coca cola news" and "algorithm" in Table X, receive larger $CS(c)$ comparing to their initial entropies $H_C(c)$, since such concepts are ambiguous in the CRN. For example, "coca cola bear" is ambiguous and thus it also refers to other ambiguous concepts, such as "coca cola product", "polar bear", and "Teddy Bears". Similarly, the concept "algorithm" also refers to a large variety of computational algorithms, such as "greedy algorithm", "divide and conquer algorithm", and "dynamic programming", and thus resulting in larger $CS(c)$. On the other hand, "dow jones index", "burberry", and "nike" concepts in Table X receive a lower $CS(c)$

comparing to their initial entropies $H_C(c)$, since these concepts retrieve commonly known information, such as well-known financial indices, well-known associations, and well-known companies with little ambiguity. Finally, "hotpot", "fantastic 4" and "dell" concepts receive $CS(c)$ similar to $H_C(c)$.

For the location Entropy Scores $LS(c)$ given in Table X, we observe that the international brands "nokia", "puma" and "new balance" are location ambiguous concepts. They receive a larger $LS(c)$ comparing to their initial location entropies $H_L(c)$. On the other hand, concepts such as "tian tan buddha", "columbia" and "frankfurt" receive lower $LS(c)$, since they refer to some specific locations that do not give rise to high location entropy. Thus, the $LS(c)$ value of such concepts also decreases compared to $H_L(c)$. Finally, the $LS(c)$ values for the famous international brands "starbacks", "pacific coffee" and "fridge" stay the same, since the H_L value for "starbuck" and "pacific coffee" are generally high, and LS values for the well-known international actor "keira knightley" remain high.

7.4. Noise Tolerance of Content and Location Scores

As discussed in Section 4.2, if a concept gains too high or low initial content entropy or location entropy due to some noisy concepts extracted from the search results, the entropies should still converge to a stable point by EntropySmooth. We now conduct experiments starting from noisy initial content entropies of the concepts retrieved by the three search engines. As shown in Table XI, H_C+5 means that a score of 5 has been imposed (added) on the initial content entropies to represent the influence of noise. Using the same notation, "$\uparrow$" means an increase of $CS(c)$ comparing to $H_C(c)$, "$\simeq$" means that little change between $CS(c)$ and $H_C(c)$, and "$\downarrow$" means a decrease of $CS(c)$ comparing to $H_C(c)$. We can see that for different classes of concepts, the noisy content Entropy Scores can still converge to values that are close to the original content Entropy Scores $CS(c)$ after 100 iterations. By imposing noise on initial entropies, the original total entropy of the CRN is changed. However, we can see that the new equilibrium point is still accurate within 3 significance figures for all the three search engines. We also performed similar experiments on the location entropy for the three search engines (cf. more detailed results are present in the online appendix [Appendix 2014]), where $H_L + n$ means that a value of n has been added or subtracted from the initial location entropies to represent influence of noise. The result implies similar conclusion as Content Scores: Location Scores are tolerant to initial noise and the EntropySmooth algorithm is effective.

7.5. Closely Linked Subgraphs in CRN

Table XIV shows that *Complete Subgraphs* in CRN can be used to effectively obtain semantically related concepts. However, due to the strict definition of complete subgraphs, which requires having complete pairwise links between all the concepts within the subgraph, the number of complete subgraphs in the result is limited, and thus giving rise to many queries with zero $CO(q, SE)$. To tackle this problem, we propose the notion of relaxed *Closely Linked Subgraphs*. We only require each concept in the subgraph is link to at least $\theta_l \times S$ other concepts, where θ_l is a relaxation threshold defined by the user and S is the total number of concepts in the subgraph.

We define a relaxed version of $CO_R(q, SE)$ as follows.

$$CO_R(q, SE) = \frac{\frac{\sum_{i=1}^{n} PC(q,c_i)}{\sum_{j=1}^{m} PC(q,c'_j)}}{MAX\_CO_R(SE)} \tag{11}$$

where $c_1, c_2, c_3, \ldots, c_n$ are the concepts within the same closely linked subgraph and $c'_1, c'_2, c'_3, \ldots, c'_m$ are the concepts connecting to q in other subgraphs. Similarly, we also propose a relaxed version of $C^2_R(q, SE)$ as follows:

$$C^2_R(q, SE) = CO_R(q, SE) \cdot (IC_C(q, SE) + IC_L(q, SE)). \tag{12}$$

Table X. Sample Concepts Categorized in the Directions of Changes in Entropies

	Concept (Google)	$H_C(c)$	$CS(c)$	Concept (Google)	$H_L(c)$	$LS(c)$
$\uparrow$	coca cola bear	6.286	8.657	mp3	6.465	8.7991
$\uparrow$	magic history	6.0917	9.4388	nokia	7.3913	10.0105
$\uparrow$	playing card	6.6474	9.0696	psp	7.0994	9.5968
$\simeq$	hotpot	5.7855	5.7895	cpu	8.0603	8.0619
$\simeq$	sushi	7.0469	7.0404	starbucks	9.1795	9.1404
$\simeq$	training dog	7.2295	7.2415	the great wall	8.2372	8.2601
$\downarrow$	dow jones index	6.3892	4.8355	empress dowager cixi	8.8564	1.8267
$\downarrow$	redcross	7.0538	4.9645	sun yat sin	10.6442	5.628
$\downarrow$	spca	7.2015	5.3829	tian tan buddha	11.0548	1.6452
	Concept (Yahoo)	$H_C(c)$	$CS(c)$	Concept (Yahoo)	$H_L(c)$	$LS(c)$
$\uparrow$	apple tree	6.4441	9.5461	fridge	8.6173	12.9399
$\uparrow$	coca cola news	7.1084	10.2908	price mobile phone	6.2328	10.3079
$\uparrow$	drama	6.3556	9.5051	puma	8.3669	12.9364
$\simeq$	classified ad	6.6196	6.6459	pacific coffee	8.5521	8.5391
$\simeq$	fantastic 4	6.7603	6.7601	seed	8.3890	8.4144
$\simeq$	focal length	6.3394	6.3406	database	9.8702	9.8787
$\downarrow$	apple pie recipe	6.2875	4.2825	christ	6.6052	2.0740
$\downarrow$	burberry	6.8057	4.7308	coca cola product	8.1846	2.1251
$\downarrow$	gold price	4.1341	1.7248	columbia	8.1653	2.6566
	Concept (Bing)	$H_C(c)$	$CS(c)$	Concept (Bing)	$H_L(c)$	$LS(c)$
$\uparrow$	algorithm	6.8611	8.7501	new balance	7.1085	9.4053
$\uparrow$	calculator	6.8161	8.5323	focal length	6.2874	8.5003
$\uparrow$	cat	6.7584	9.0577	lens	6.5605	8.8052
$\simeq$	dell	7.3485	7.3490	fridge	8.5642	8.5358
$\simeq$	hp	7.2108	7.2311	keira knightley	7.7418	7.7268
$\simeq$	morning star	6.8241	6.8491	keroro wallpaper	7.7544	7.7609
$\downarrow$	apple	2.7898	0.6614	frankfurt	7.8909	1.7100
$\downarrow$	nike	5.2728	3.1618	redcross	5.7143	0.7179
$\downarrow$	red eye	4.0940	1.6372	public housing	7.5063	2.6511

Table XI. Effects to $CS(c)$ by Adding Noise to $H_C(c)$

	Concept (Google)	$H_C + 5$	CS'
$\uparrow$	coca cola bear	11.2860	8.6578
$\uparrow$	magic history	11.0917	9.4396
$\uparrow$	playing card	11.6474	9.0704
$\simeq$	hotpot	10.7855	5.7901
$\simeq$	sushi	12.0469	7.0410
$\simeq$	training dog	12.2295	7.2422
$\downarrow$	dow jones index	11.3892	4.8359
$\downarrow$	redcross	12.0538	4.9650
$\downarrow$	spca	12.2015	5.3834
Category	Concept (Yahoo)	$H_C + 5$	CS'
$\uparrow$	apple tree	11.4441	9.5467
$\uparrow$	coca cola news	12.1084	10.2915
$\uparrow$	drama	11.3556	9.5057
$\simeq$	classified ad	11.6196	6.8144
$\simeq$	fantastic 4	11.7603	6.9285
$\simeq$	focal length	11.3394	6.5093
$\downarrow$	apple pie recipe	11.2875	4.2828
$\downarrow$	burberry	11.8057	4.7311
$\downarrow$	gold price	9.1341	1.7249
Category	Concept (Bing)	$H_C + 5$	CS'
$\uparrow$	algorithm	11.8611	8.7507
$\uparrow$	calculator	11.8161	8.5329
$\uparrow$	cat	11.7584	9.0583
$\simeq$	dell	12.3485	7.4914
$\simeq$	hp	12.2108	7.3735
$\simeq$	morning star	11.8241	6.9914
$\downarrow$	apple	7.7898	0.6614
$\downarrow$	nike	10.2728	3.1620
$\downarrow$	red eye	9.0940	1.6373

Table XII presents the statistics of the subgraphs obtained when θ varies from 0.9 to 0.5. The results of the C^2_R are presented in Table XIII. One observation is that $C^2_R(q, SE)$ scores are much larger than $C^2(q, SE)$, indicating that more queries are treated as semantically related with this metric. Another observation is that the ranking of the three search engines based on C^2_R is the same as that obtained by C^2: Google yields the highest $C^2_R(q, SE)$, while Yahoo yields the lowest $C^2_R(q, SE)$.

We also observe that the closely linked subgraphs from Google(0.9) are exactly the same as the complete ones. As for Yahoo, the subgraphs from Yahoo(0.9) and Yahoo(0.8) are the same as the complete counterparts. This shows that Yahoo CRN is more sparsely connected comparing to Google CRN. Table XIV shows some example closely linked subgraphs with different relaxation thresholds θ_l. We observe that when θ_l is high, closely linked subgraph can retrieve more semantically relevant concepts (e.g., "hair growth" in Google(0.8) and "people" in Bing(0.8)) compared to the complete counterparts. However, if θ_l is too low, some noisy and irrelevant concepts (e.g., "directory" in Bing(0.5)) may be included in subgraphs.

7.6. Predicting Query Terms via CRN

We now evaluate the effectiveness of Algorithm 1 in determining semantically relevant concepts from CRN. In this Section, we focus on evaluating whether these semantically related concepts for an input query appears as more refined queries in the same search session. In the evaluation, we utilize the taxonomy of query reformulation proposed in [Huang and Efthimiadis 2009] to segment raw search query log into search sessions. The goal of deriving search sessions is to group consecutive and semantically related queries together. In our experiment, we sample a subset of the AOL dataset and group the queries into 23,236 search sessions. By using the input terms in the first query

Table XII. Statistics of Closely Linked Subgraphs

	# Subgraphs	Max Size	Avg Size
Google(0.9)	978	7	3.19
Yahoo(0.9)	159	4	3.04
Bing(0.9)	196	6	3.09
Google(0.8)	980	8	3.20
Yahoo(0.8)	159	4	3.04
Bing(0.8)	196	7	3.10
Google(0.7)	966	9	3.25
Yahoo(0.7)	159	5	3.04
Bing(0.7)	194	8	3.12
Google(0.6)	908	9	3.82
Yahoo(0.6)	157	6	3.31
Bing(0.6)	193	8	3.49
Google(0.5)	2895	15	3.61
Yahoo(0.5)	1073	6	3.17
Bing(0.5)	1190	11	3.26

Table XIII. C_R^2 Analysis (Average)

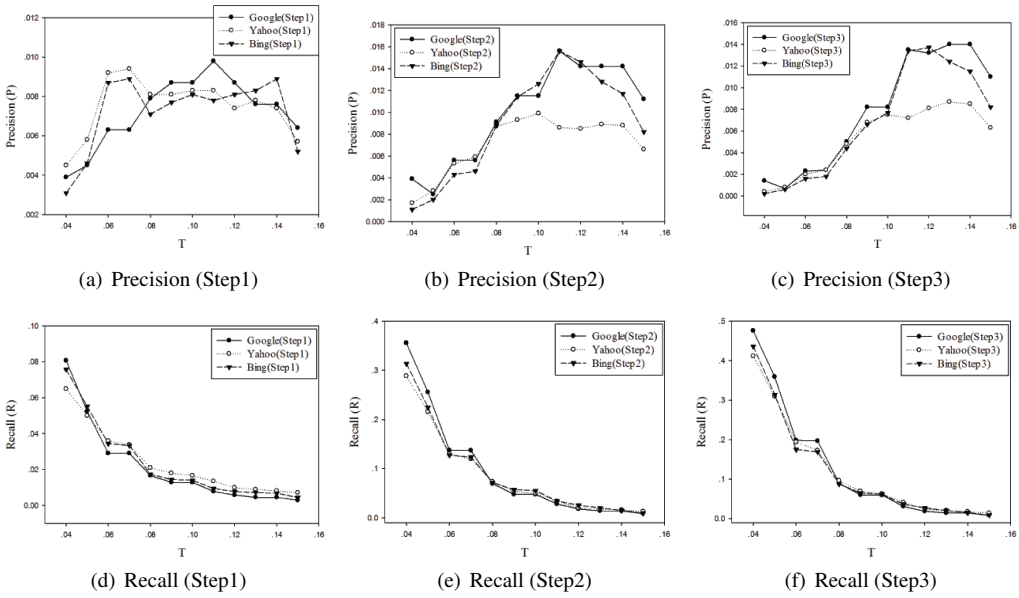
Search Engine	Level 0	Level 1	Level 2	Overall
Google(0.9)	0.11352	0.09926	0.02409	0.03266
Yahoo(0.9)	0.01696	0.01755	0.00187	0.00364
Bing(0.9)	0.06438	0.02714	0.00260	0.00556
Google(0.8)	0.11352	0.10015	0.02438	0.03302
Yahoo(0.8)	0.01696	0.01755	0.00187	0.00364
Bing(0.8)	0.06438	0.02759	0.00260	0.00561
Google(0.7)	0.11410	0.10469	0.02578	0.03476
Yahoo(0.7)	0.01696	0.01760	0.00188	0.00366
Bing(0.7)	0.06890	0.02821	0.00260	0.00570
Google(0.6)	0.14208	0.13426	0.03473	0.04602
Yahoo(0.6)	0.02044	0.02041	0.00218	0.00424
Bing(0.6)	0.08414	0.03868	0.00298	0.00724
Google(0.5)	0.33487	0.27929	0.07580	0.09911
Yahoo(0.5)	0.12374	0.07584	0.00935	0.01712
Bing(0.5)	0.23959	0.12376	0.01311	0.02623

Table XIV. Complete Subgraphs and Closely Linked Subgraphs Comparison(Concepts in bold only exist in closely linked subgraph

Search Engine	Complete Subgraph	Closely Linked Subgraph
Google(0.9)	*The same as Closely Linked Subgraph*	*The same as Complete Subgraph*
Google(0.8)	baldness, hair loss, hair replacement	baldness, **hair growth**, hair loss, hair replacement
Google(0.7)	blood center, blood donation, blood service	blood center, blood donation, blood service, **give blood**
Google(0.6)	air rifle, airgun, bb gun	**air gun**, air rifle, airgun, bb gun, **rifle pistol**
Google(0.5)	gov, office, united state	**federal**, gov, office, **regulation**, united state
Yahoo(0.9)	*The same as Closely Linked Subgraph*	*The same as Complete Subgraph*
Yahoo(0.8)	*The same as Closely Linked Subgraph*	*The same as Complete Subgraph*
Yahoo(0.7)	address, find people, people search	address, find people, **lookup**, people search
Yahoo(0.6)	agent, real estate, realtor	agent, **listing**, **mls**, real estate, realtor
Yahoo(0.5)	bed, futon, mattress	bed, futon, mattress, **sofa**
Bing(0.9)	*The same as Closely Linked Subgraph*	*The same as Complete Subgraph*
Bing(0.8)	blog, community, friend, make, photo, share	blog, community, friend, make, **people**, photo, share
Bing(0.7)	find, free, search, shopping	find, free, search, shopping, **tool**, **yahoo**
Bing(0.6)	allah, islamic, muslim	allah, islamic, **muhammad**, muslim, **qur**
Bing(0.5)	animal, dog, pet	animal, **directory**, dog, **grooming**, pet

of a session into Algorithm 1, we evaluate whether the obtained terms in CRN can effectively cover the terms of the later queries in the same session. Given a search session $(q_1, q_2, ..., q_n)$, G_q denotes a set of the terms in $(q_2, q_3, ..., q_n)$. We utilize Algorithm 1 to obtain the terms, denoted as G_c, and adopt standard *Precision* (P) and *Recall* (R) to evaluate the effectiveness of the retrieved concepts. We compute the two metrics with respect to different steps of walk on the three CRNs.

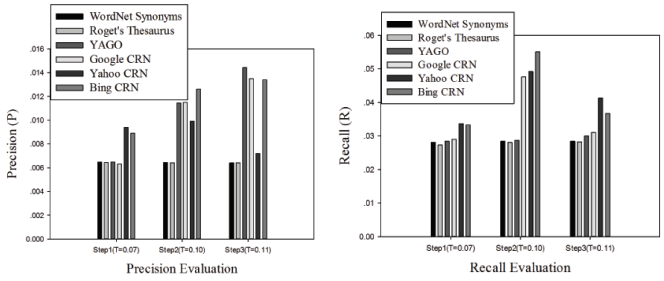
Since the link weight of the parent-child relationship in CRN is generally between 0.01 and 0.15, we set a threshold T on the link weight $Link(t, c)$, ranging from 0.01 to 0.15 in Algorithm 1. We carry out at most 3 steps of walk on each CRN, since the $4th$ step retrieves too many irrelevant concepts and thus heavily reduces the coherency of the retrieved concepts. The results of average P and average R of the three CRNs are shown in Figure 6. For P, we observe that Yahoo(Step1) yields the highest quality when the T is low. On the other hand, when T is high, Google(Step2), Bing(Step2) and Google(Step3) yield the best results. For R, Google CRN demonstrates the best performance in each step. This shows that Google CRN can effectively retrieve semantically related concepts that are very likely to be submitted by the users in later searching. Google CRN yields better results, since Google CRN is larger in size and the concept connection in Google CRN is much tighter than Yahoo and Bing CRNs. This also shows that the proposed $C^2(q, SE)$ provides accurate estimation of the performance of a search engine.



(a) Precision (Step1) (b) Precision (Step2) (c) Precision (Step3)

(d) Recall (Step1) (e) Recall (Step2) (f) Recall (Step3)

Fig. 6. Precision and Recall of CRN Query Suggestion



(a) Precision (b) Recall

Fig. 7. Query Suggestion Comparison

As for the baselines, we conduct three steps of walk on WordNet, Roget's Thesaurus and YAGO for the same dataset to compute P and R values. Since the meaning of links in these semantic networks is quite different from that of CRN, we do not set any threshold on the link weight. Essentially, we conduct unweighted walk on these semantic networks and employ all the reachable words to calculate P and R values. We consider the following six relationship linkages in WordNet: *Synonyms*, *Holonyms*, *Hypernyms*, *Hyponyms*, *Meronyms*, and *All* that is the combination of all the previous five relations. We observe that the results of *Holonyms*, *Hypernyms*, *Hyponyms*, *Meronyms* and *All* are much worse than that of CRN. Thus, we focus on the comparison of CRN with WordNet's *Synonyms*. We present the results in Figure 7. By employing a broad coverage, unrestricted and up-to-date vocabulary mined from the large number web pages, which are constantly crawled and updated, CRN can successfully discover a larger set of semantically related terms for query suggestion, and thus yielding higher P and R compared to the baselines.

7.6.1. Entropy Filtering of Undesirable Concepts. We also evaluate the effectiveness of adapting H_C, H_L, $H_C + H_L$, CS, LS and $CS + LS$ for filtering out undesirable concepts retrieved from the random walk algorithm. We refer undesirable concepts to mean ambiguous concepts with high

entropies. Intuitively, they are general terms with broad meanings, and can be easily reached by the random walk algorithm due to their high degree of connectivity in CRN. We believe that these highly ambiguous concepts are likely to be false positives in the inferred clusters. Thus, we propose to filter out concepts with high entropies after obtaining all the suggestion candidates using 3 steps of walk in Algorithm 1. After obtaining all the suggestion candidates by Algorithm 1, we filter out 10 candidates with the highest entropies ($Top10$), and 10 candidates with the highest entropies ($Bottom10$). We compare $Top10$ with $Bottom10$, and evaluate their effectiveness in the query suggestion task using standard *Precision* (P) and *Recall* (R). The results are shown in Table XV. We observe that the initial entropies, H_C, H_L and $H_C + H_L$, are not very effective in distinguishing undesirable concepts. This is because some concepts may gain much higher or lower initial entropies due to the inclusion of some noisy concepts extracted from the search results as discussed in Section 7.4. However, once the entropies converge to a stable point as entropy scores, CS, LS and $CS+LS$, can effectively filter out undesirable concepts in the suggestion candidates. Both the P and R values can significantly be improved by using the $Bottom10$ instead of the $Top10$ candidates, which are possibly general or ambiguous terms. With $CS + LS$ and Google as the backend search engine, the R value is significantly improved from $R_{GoogleTop10} = 0.0530$ to $R_{GoogleBottom10} = 0.0966$. This shows that entropy is an important measure that gauges the quality of concepts. Thus, it is effective to make use the entropy to filter out undesirable concepts in the query suggestion task.

Table XV. Entropy Filtering Evaluation

	Search Engine	$P(Top10)$	$R(Top10)$	$P(Bottom10)$	$R(Bottom10)$
H_C	Google	0.0006	0.1123	0.0004	0.0394
	Yahoo	0.0011	0.1616	0.0012	0.1789
	Bing	0.0011	0.1221	0.0010	0.0998
	Search Engine	$P(Top10)$	$R(Top10)$	$P(Bottom10)$	$R(Bottom10)$
H_L	Google	0.0005	0.0510	0.0005	0.0918
	Yahoo	0.0013	0.1115	0.0011	0.1764
	Bing	0.001	0.0727	0.0008	0.0868
	Search Engine	$P(Top10)$	$R(Top10)$	$P(Bottom10)$	$R(Bottom10)$
$H_C + H_L$	Google	0.0006	0.0707	0.0006	0.0753
	Yahoo	0.0013	0.1269	0.0012	0.1769
	Bing	0.0012	0.0851	0.0009	0.0872
	Search Engine	$P(Top10)$	$R(Top10)$	$P(Bottom10)$	$R(Bottom10)$
CS	Google	0.0006	0.0530	0.0006	0.0958
	Yahoo	0.0010	0.0938	0.0012	0.1840
	Bing	0.0006	0.0530	0.0006	0.0958
	Search Engine	$P(Top10)$	$R(Top10)$	$P(Bottom10)$	$R(Bottom10)$
LS	Google	0.0006	0.0503	0.0006	0.0957
	Yahoo	0.0009	0.0885	0.0012	0.1841
	Bing	0.0006	0.0503	0.0006	0.0957
	Search Engine	$P(Top10)$	$R(Top10)$	$P(Bottom10)$	$R(Bottom10)$
$CS + LS$	Google	0.0006	0.0516	0.0006	0.0966
	Yahoo	0.0010	0.0923	0.0012	0.1841
	Bing	0.0006	0.0516	0.0006	0.0966

8. CONCLUSIONS

In this paper, we propose a methodology to construct a new semantic network, *Concept Relation Network (CRN)*, which tackles the uncertainty and dynamics of web content. CRN can be automatically constructed and maintained using commonly used search engines. Thus, the new semantic network is feasible and practical in real-life web applications. We propose two important applications for CRN: 1) search engine and web analytic and 2) semantic information retrieval. Throughout the paper, we demonstrate that CRN is an effective semantic network to gain deeper insight of search engines in terms of information coverage and diversity.

For future work, we observe that CRNs constructed at different time and location are going to be different. Thus, comparing the context of a CRN snapshot constructed at a particular time and location is an interesting and is worth further exploration. In addition, CRN can also be constructed based on a user's search history as a form of search engine user profile, which is useful in enhancing personalization systems. Finally, our approach of using CRN also paves the way to include more sophisticated relationships (e.g. is-a, has-a, member-of, antonum, etc) in the analysis. It is thus an interesting issue to study different combinations of paths and connections between the concepts to enrich the context of CRN.

REFERENCES

Appendix 2014. (2014). http://www.cse.ust.hk/~kwtleung/appendix.pdf.

Abdur Chowdhury and Ian Soboroff. 2002. Automatic Evaluation of World Wide Web Search Services. In *Proc. of the SIGIR Conference*.

Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. 1991. Using statistics in lexical analysis. In *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Jianfeng Gao, Gu Xu, and Jinxi Xu. 2013. Query Expansion Using Path-Constrained Random Walks. In *Proc. of the SIGIR Conference*.

Geographic Names for Geopolitical Areas from GNS 2013. (2013). http://earth-info.nga.mil/gns/html/namefiles.htm.

J. Huang and E. N. Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In *Proc. of the CIKM Conference*.

Alpa Jain, Umut Ozertem, and Emre Velipasaoglu. 2011. Synthesizing high utility suggestions for rare web search queries. In *Proc. of the SIGIR Conference*.

Steve Lawrence and C. Lee Giles. 1998. Searching the world wide web. *Science* (1998).

Steve Lawrence and C. Lee Giles. 2000. Accessibility of information on the Web. *Intelligence* (2000).

Kenneth Wai-Ting Leung, Hing Yuet Fung, and Dik Lun Lee. 2011a. Constructing Concept Relation Network and Its Application to Personalized Web Search. In *Proc. of the EDBT Conference*.

Kenneth Wai-Ting Leung, Dik Lun Lee, Wilfred Ng, and Hing Yuet Fung. 2011b. A Framework for Personalizing Web Search with Concept-Based User Profiles. *ACM TOIT* (2011).

Kenneth Wai-Ting Leung, Wilfred Ng, and Dik Lun Lee. 2008. Personalized Concept-Based Clustering of Search Engine Queries. *IEEE TKDE* (2008).

Zhen Liao, Daxin Jiang, Enhong Chen, Jian Pei, Huanhuan Cao, and Hang Li. 2011. Mining Concept Sequences from Large-Scale Search Logs for Context-Aware Query Suggestion. *ACM TSIT* (2011).

H Liu and P Singh. 2004. Focusing on ConceptNet's natural language knowledge representation. In *Proc. of the KES Conference*.

Jose G. Moreno, Gaël Dias, and Guillaume Cleuziou. 2014. Query Log Driven Web Search Results Clustering. In *Proc. of the SIGIR Conference*.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. The PageRank citation ranking: Bringing order to the Web. *Technique Report, Computer Science Department, Stanford University* (1999).

Roget's Thesaurus 2013. (2013). http://thesaurus.com/Roget-Alpha-Index.html.

Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge - Unifying WordNet and Wikipedia. In *Proc. of the WWW Conference*.

Jan Vosecky, Kenneth Wai-Ting Leung, and Wilfred Ng. 2014. Collaborative Personalized Twitter Search with Topic-language Models. In *Proc. of the SIGIR Conference*.

World Gazetteer 2013. http://www.world-gazetteer.com/wg.php?x=1129163518&men=stdl&lng=en&gln=xx&dat=32&srt=npan&col=aohdq. (2013).

Yabo Xu, Ke Wang, Benyu Zhang, and Zheng Chen. 2007. Privacy–enhancing personalized web search. In *Proc. of the WWW Conference*.

Lina Yao, Quan Z. Sheng, Anne H. H. Ngu, Helen Ashman, and Xue Li. 2014. Exploring recommendations in internet of things. In *Proc. of the SIGIR Conference*.

Majid Yazdani and Andrei Popescu-Belis. 2013. Computing Text Semantic Relatedness Using the Contents and Links of a Hypertext Encyclopedia. *Artificial Intelligence* 194 (Jan. 2013), 176–202. DOI:http://dx.doi.org/10.1016/j.artint.2012.06.004