Mining Enterprise Websites for Association Thesaurus Construction

Luciano Barbosa
AT&T Labs – Research
180 Park Ave
Florham Park, NJ 07932
lbarbosa@research.att.com

ABSTRACT

Enterprise websites are useful resources for obtaining information about products and services of companies. Typically on these websites, a product is associated to a Web page, and related products are connected by hyperlinks. As a result, the connectivity graph of an enterprise website exposes the company's products (nodes) and how they are associated (links). This paper presents a novel approach that mines these graphs in order to build association thesauri for enterprises. An association thesaurus represents implicit associations between concepts (company-related information in the context of this work). To perform this task, our approach first executes a breadth-search crawl in the website, building an initial thesaurus. Next, it employs probabilistic modelling to remove non-relevant content and assign weights to the associations in the thesaurus. We evaluated the association thesauri produced by our technique in the query suggestion scenario. We measured the quality and the diversity of the suggestions obtained from the thesauri, and compared it against suggestions from two commercial search engines. The results show that in both aspects our method obtained better performance.

1. INTRODUCTION

Thesauri have been widely used in many areas such as information retrieval[12], and natural language processing[13]. A thesaurus represents a graph association between concepts or words. These associations can be explicit, e.g., hyponym and hypernym, or implicit, e.g., words are associated simply because they co-occur. Thesauri with implicit associations are called association thesauri.

In this paper, we focus on automatically building *association thesauri for enterprises*. Enterprise-based thesauri are useful resources for business-centric applications in general. For instance, existing social media analytics tools for business (e.g. Sysomos[1] and Attensity[2]) could use thesauri to extract information associated to products and services of companies [4]. Sites and tools specialized in enterprise content search (e.g., BizReport[3] and Northen Light[4]) would benefit from an enterprise thesaurus to provide suggestions for issued queries.

Previous works have proposed automatic methods to construct domain-specific thesauri based on the collocation of words in a text [7] or the co-occurrence of links in a page [6]. Similar to [6], our approach also looks at the link structure to build thesauri, but instead of using collocation, it relies on the graph connectivity of the website. Typically on these websites, a product is associated to a Web page, and related products are connected by hyperlinks. As a result, the connectivity graph of an enterprise website exposes the company's products (nodes) and how they are associated (links). In addition, these websites present some kind of hierarchical structure where pages close to its root represent more generic products whereas deeper pages more specific ones.

In this work, we exploit these characteristics to build association thesauri for enterprises. First, we execute a breadth-search crawl in the website, building an initial hierarchical thesaurus. Next, we use mutual information to remove non-relevant content from this thesaurus (e.g., "site map" and "contact us" pages). Finally, we employ probabilistic modelling to assign weights to the associations, since a concept/node is not uniformly associated to its neighbours.

The remainder of the paper is organized as follows. In Section 2, we introduce our method of mining enterprises' websites to obtain query suggestions. Section 3 presents the experimental evaluation that assesses different aspects of our solution and compares it against query suggestions from two commercial search engines. Related work is presented in Section 4 and conclusions and future work in Section 5.

2. THESAURUS CONSTRUCTION

In this section, we describe the steps performed by our approach to mine association thesaurus from companies' websites.

Sixteenth International Workshop on the Web and Databases (WebDB 2013), June 23, 2013 - New York, NY, USA.

[1] http://www.sysomos.com/
[2] http://www.attensity.com/
[3] http://www.bizreport.com/
[4] http://northernlight.com/

2.1 Building the Graph

In enterprise websites, pages are usually associated to products, and related products and services are connected by hyperlinks in a hierarchical way. Consider, for instance, the two Web pages from the AT&T website presented in Figure 1. The initial page of the website contains links to top-level products of the company such as wireless and digital tv, and the wireless page points to more specific products (e.g. free phones and smartphones). Based on that, our approach builds the thesaurus graph structure as follows. First, it performs a breadth-first search crawl in the website. Each visited page corresponds to a node in the thesaurus, and a hyperlink represents an association between nodes. To exploit the hierarchical structure of the site, nodes are only linked to their children. The root node corresponds to the the website's initial page, and its label is the company name. The label of an internal node n is the anchor text used by its parent node to point to n. In its raw form, however, anchor texts are very noisy (e.g., "get downloads" or "log in to pay bill"). To obtain a more meaningful and concise representation of the anchors, we extract the noun phrases from them using a PCFG parser[5] removing articles, adjectives, adverbs, pronouns and conjunctions from the noun phrases. Figure 2 presents an example of a small portion of a thesaurus extracted from the AT&T website using the described strategy. As one can see, most of the nodes are products or services of the company ("u-verse", "digital tv") or some information related to its products ("customer"), and connected nodes have some type of relation, e.g., "att directv" and "channel lineup" are associated to "digital tv" .

2.2 Weighting the Connections

The next step in building an association thesaurus is to weight the relationships between nodes. To motivate that, we refer again to the graph in Figure 2. One can say that the node "free phones", for instance, is more associated to "samsung a777", which is a free phone offered by AT&T, than the node "customer". To assign weights to connections in this graph, our strategy deals with two different types of nodes: nodes in the first layer of the thesaurus, which are top-level products, and nodes in the deeper layers, which are commonly associated to specific products. These different categories of nodes can be clearly seen in the thesaurus in Figure 2. Nodes in the first layer such as "wireless" and "digital tv" represent top-level products, and deeper nodes e.g. "gps" and "netbooks" are more specific products/services. For each one of these categories, we implement a different weighting strategy.

2.2.1 First Layer

The connections in the first layer of the thesaurus are composed by the links from the root page of the Website (root node). These links usually point to three different types of pages: (1) top-level products; (2) specific products; or (3) site-specific sections. Consider again the first page of the AT&T website presented in Figure 1. It contains links to top-level products of the company (e.g. wireless, digital tv and internet), to specific products (e.g. gophone), and to site-specific pages (e.g. privacy policy and terms of use). To embed some notion of abstraction in the thesaurus, for the first layer we are primarily interested in top-level products.

Company's name	Anchor	PMI-IR
apple	iphone	0.069
apple	environment	0.00014
motorola	motoblur	0.04
motorola	news	0.00005

Table 1: Examples of PMI-IR values.

To remove site-specific links from the first layer, our algorithm verifies whether the children of the root node r has a significant association with r using pointwise mutual information (PMI) [13]. PMI is a common strategy to calculate association between words. Thus, to measure the PMI between r in a given child c, we use their labels. More precisely:

$$PMI(Label_r, Label_c) = log\frac{p(Label_r, Label_c)}{p(Label_r)p(Label_c)} \quad (1)$$

where $Label_r$ is the root node's label (company's name), and $Label_c$ is the label (anchor text) of a root's child node. Since $p(Label_r)$ is common for all children and removing the log, we have the conditional probability:

$$p(Label_r|Label_c) = \frac{p(Label_r, Label_c)}{p(Label_c)} \quad (2)$$

This conditional probability can be estimated using maximum likelihood from a corpus. Similar to previous approaches [16, 14, 21], we use a search engine index as a corpus. This way of calculating associations between words is known in the literature as pointwise mutual information using information retrieval (PMI-IR) [20]. Thus, PMI-IR is simply:

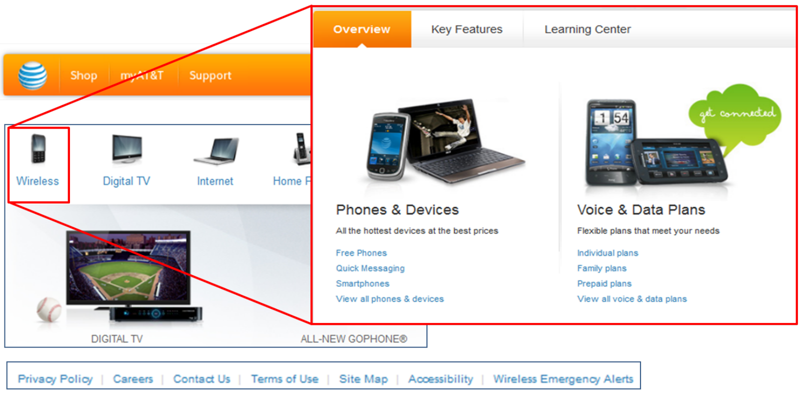
$$PMI-IR(Label_r, Label_c) = \frac{|HITS(Label_r, Label_c))|}{|HITS(Label_c)|} \quad (3)$$

where HITS is the function that returns the number of documents from a search engine query. Table 1 illustrates some examples of the PMI-IR scores using counts from Google. One can see, for instance, "iphone" has a higher PMI-IR score for "apple" than "environment", and "motoblur" (motoblur is a phone device sold by Motorola) has a higher PMI-IR score for "motorola" than "news". Using this technique, our algorithm considers only children nodes of the root node whose PMI-IR is score higher than a certain value[6].

After pruning nodes that are not associated to the root node, the next step is to give higher weights to nodes that represent top-level products in contrast to specific ones. To do that, our approach exploits the fact that, in enterprise websites, links to top-level products' pages are available in navigation bars to facilitate the access to them. As a result, a random walker in the Website graph has a greater chance to visit a top-level product's page than a specific one. Based on that, we model the Website graph as a markov chain and we use the PageRank algorithm [15] (dumping factor

[5] In this work, we used the Stanford Parser [9]

[6] We empirically set this threshold equals to 0.001 in our experiments.



Phones & Devices

All the hottest devices at the best prices

Free Phones
Quick Messaging
Smartphones
View all phones & devices

Voice & Data Plans

Flexible plans that meet your needs

Individual plans
Family plans
Prepaid plans
View all voice & data plans

Privacy Policy | Careers | Contact Us | Terms of Use | Site Map | Accessibility | Wireless Emergency Alerts

Figure 1: Initial and wireless page of the AT&T website.

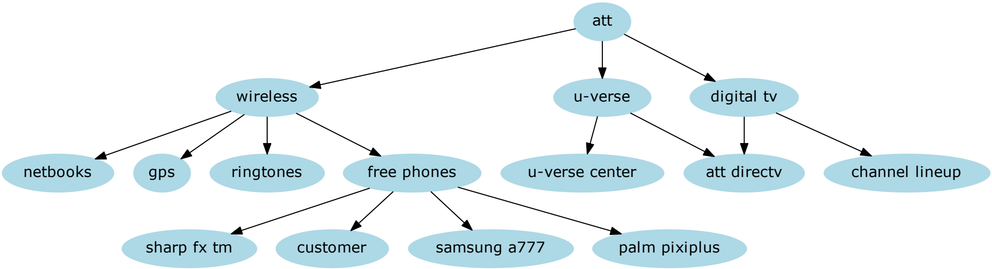


Figure 2: Portion of an unweighed thesaurus extracted from AT&T website using our strategy.

equals to 0.85) to calculate the probabilities of stationary states. These probabilities represent the weights assigned to the connections of the root node. Table 2 presents the top-8 nodes ranked using this strategy for 5 different companies. As one can see, most of the nodes are top-level products of those companies.

2.2.2 Deeper Layers

Internal nodes in the thesaurus primarily represent specific products of an enterprise. Back to the AT&T thesaurus in Figure 2, examples of internal nodes are: "gps", "samsung a777" and "att directv". Given an internal node, we aim to assign weights to its children based on how associated they are. We use probabilistic modelling to formulate this association. More specifically, we want to calculate the posterior probability:

$$p(Child_i|Node) = \frac{p(Node|Child_i)p(Child_i)}{\sum p(Node|Child_i)p(Child_i)} \qquad (4)$$

In order to calculate the likelihood $p(Node|Child_i)$, we use the distribution of the text in the anchor associated to $Child_i$. More specifically, let $anchor_i$ be the anchor text associated to $Child_i$ in Node N, S_N be the set of anchor texts in N, and S_{ALL} be the set of all anchor texts in the entire website, Thus:

$$p(Node|Child_i) = \frac{\#anchor_i \ in \ S_N}{\#anchor_i \ in \ S_{ALL}} \qquad (5)$$

The final term in the posterior probability equation (Equation 4) that we need to estimate is the prior $p(Child_i)$. A simple approach would be to assume that initially the association between a node and all its children is the same. Instead of doing that, we assume that links (URLs) of relevant children are more similar to each other than non-relevant ones. Thus, grouping these links by similarity, children nodes in the bigger clusters would be considered more important to $Node$ than the ones in smaller groups. The prior is then calculated as:

$$p(Child_i) = \frac{|Cluster'_j|}{\sum |Cluster_j|} \qquad (6)$$

where $Child_i \in Cluster'_j$, and $|Cluster|$ is the number of elements of a cluster. For clustering the URLs, we used a hierarchical clustering algorithm. The URLs are split in tokens, and the tokens are used as features. Similarity between elements is calculated using cosine distance. The clustering algorithm stops when the number of clusters is 30% of the initial number of clusters.

Company's name	Top-8 nodes
samsung	laptops,monitors,chromebook,printers,projectors, memory storage,cell phones, tvs
motorola	mobile phones,smartphones,bluetooth headsets,motoblur, android tablets,batteries,tablet accessories,software applications
t-mobile	phones devices,services,phones,mobile broadband, android phones,memory cards,chargers,batteries
at&t	u-verse bundles,data plans,individual plans, family plans,dial-up,digital tv,wireless,home phone
apple	itunes,apple support,apple hot news,iphone, mac,ipad,ipod,apple store

Table 2: Top-8 children nodes of the root node.

Table 3 presents the top-5 children of the 5 nodes obtained from the weighting process of internal nodes. Most of them have some relationship with their parent. For instance, "350 series led" is a type of Samsung monitor.

3. EXPERIMENTAL SECTION

As we mentioned before, an association thesaurus could be used in different applications (e.g. information extraction, query suggestion etc). In this section, we show some results of using the association thesaurus created by our solution in the domain of query suggestion. We evaluate the suggestions provided by our approach in terms of quality and diversity, and compare it against two commercial search engines.

3.1 Experimental Setup

Mined Websites. We mined 5 websites to extract their thesauri: Apple, AT&T, Motorola, Samsung and T-Mobile. To compose the queries, we consider the label L of a node as the original query and the combination L with the label of its children as suggestions. For instance, for the query 'samsung', the recommended queries were: 'samsung chromebook', 'samsung tvs', 'samsung cell phones', 'samsung projectors' and 'samsung printers'. We generated suggestions for 38 queries.

Baseline Methods. We compared the suggestions generated by our approach against suggestions of two commercial search engines: Bing and Google. Although these search engines are not tuned for enterprise queries, they rely on a huge amount of query logs to perform this task, whereas we just use the websites' content. Thus, for all 38 queries of our evaluation, we collected the refinements suggested by these search engines for comparison. In the remaining of this section, we use Website Miner to refer to our approach.

Evaluation Setup. To evaluate the different approaches, we performed 2 different user studies. We posted these studies on Mechanical Turk (MTurk) and asked turkers (MTurk's workers) to assess them. These were the two studies:

- Scoring suggestions: Each item in this task contains a pair of queries: original query and suggested query. Similar to previous work [11], we asked labelers to score a suggestion in a range of 1-5. A score of 5 means a very related suggestion to the original query and a score of 1 no relation at all. In addition to that, in order to justify their scores, we asked the turkers to select one of the 3 categories: the suggested query is (1) highly related to the original one and helpful, or (2) highly related to the original one and but there are better ones, or (3) not related to the query. If a turker was not familiar with some queries, we asked them to post these queries in a search engine. Each HIT (HIT is how an individual task is called on MTurk) was composed by 8 pairs of queries and we payed $0.05 per each. Since redundancy is important to mitigate the work of bad labelers, We asked 9 labels from different turkers for each HIT. A total of 300 pairs of queries were posted. The approaches were evaluated based on their overall average score;
- Diversity evaluation: for this task, we asked turkers to evaluate a set of recommendations for a given query in three categories: the set of queries is (1) comprehensive and diverse, (2) either not comprehensive or not diverse, (3) not comprehensive and not diverse. We stated in the instruction that a comprehensive set means a set that covers various aspects of the original query, and a diverse set means a set that the results of the individual queries have small overlaps. The reward for each HIT was $0.01 and, as in the previous task, we asked 9 labels from different turkers;

3.2 Results

Figure 3 shows the average scores obtained by the three approaches for the first study. The highest value, 3.75, was obtained by our approach (Website Miner), followed by Bing (3.64) and Google (3.62). For both comparisons (Google vs. Site Mining, and Bing vs. Site Mining), we ran t-tests, which showed that the mean score of our approach is significant larger than the mean of each competitor with 95% of confidence interval. Here are some examples of suggestions that obtained low scores: for Bing, "dell tablet" for the query "samsung tablets"; for Google, "dell laptops" for the query "samsung laptops"; and for our approach, 'att digital tv" for "att".

By looking at the justification given by the turkers about their scores (see Figure 4), we observe that our approach was able to provide a higher number of related recommendations

Node	Top-5 children nodes
samsung monitors	350 series led,23.6 lcd monitor, samsung central station,hdtv led monitor
iphone	apps,ios developers,tips tricks, ios 4,business features
motorola mobile phones	milestone x,i576,motorola quantico, droid pro,i886
at&t wireless	messaging data,smartphones pdas,additional phones, refurb cell phones, prepaid plans
t-mobile prepaid plans	android-powered phones,comet black refurbished, nokia x2, optimus t google, samsung t359

Table 3: Examples of top-5 children nodes for internal nodes.

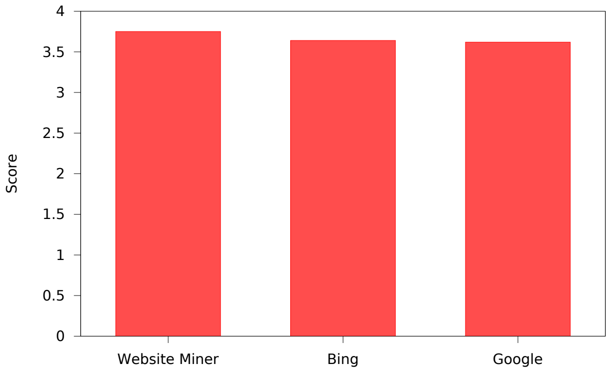


Figure 3: Average quality scores for each approach.

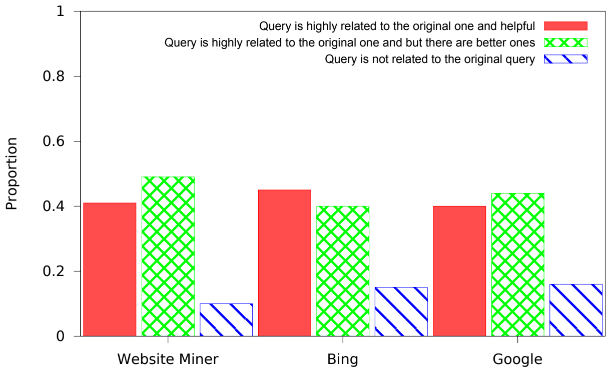


Figure 4: Justification given by the turkers about their scores.

to the original query than the other approaches. Only 10% of the recommended queries were classified as not related to the original query whereas Bing obtained 15% and Google 16%.

Another aspect that we investigated was the diversity produced by the three approaches. Figure 5 depicts the proportion of sets of queries in the diversity categories for each approach. The results show that our method produces more diverse and comprehensive result sets than the other ones. 74% of our queries were classified as 'comprehensive and diverse', whereas 70% from Bing and 67% from Google. The good performance of our approach comes from the fact that the page of a product usually tries to expose many different aspects of it.

4. RELATED WORK

Traditionally, automatic thesaurus are created by exploiting word collocation in large document collections [17, 7]. For instance, Qiu and Frei [17] proposes a query expansion approach based on statistical co-occurrence data in a large document set. Although effective, they are computationally expensive and require a large document collections in the domain. Instead of looking at word collocation, Ito et al [6] proposed a method to create an association thesaurus from Wikipedia using link co-occurrence. Two links co-occur if they appear in the same article within a certain distance of each other. Although we also use links to create the enterprise thesaurus, our approach exploits the graph connectivity to make associations instead of link collocations. Chen et al. [3] presents a different approach that mines anchors texts from shopping websites (as e.g. ebay) for query expansion. This work has some similarities with ours but they do not focus specifically on enterprise-based search.

Regarding query suggestions, a common strategy is the usage of query logs [2, 5, 8] by mining related queries posted by users. The main issues with this technique is that it requires access to a huge volume of queries from previous searches to mine reformulations; and the time to obtain the reformulations depends on the search engine traffic. Our approach, on the other hand, does not have these requirements: suggestions can be presented once they were mined from the website. Similar to our approach, Kraft and Zien [10] showed that query suggestions from anchor texts are useful and complementary to the ones from query logs. Other query suggestion techniques use the result pages of a given query to suggest refinements for it [1, 12, 11]. Since they heavily rely on the content of the result pages, if these results were not relevant to the posted query, they would produce poor recommendations. In addition, they can be computationally expensive [18]. Our approach is agnostic regarding the results returned by the query.

5. CONCLUSIONS

We have presented a new strategy to build association thesauri for enterprises. The approach relies on the observa-

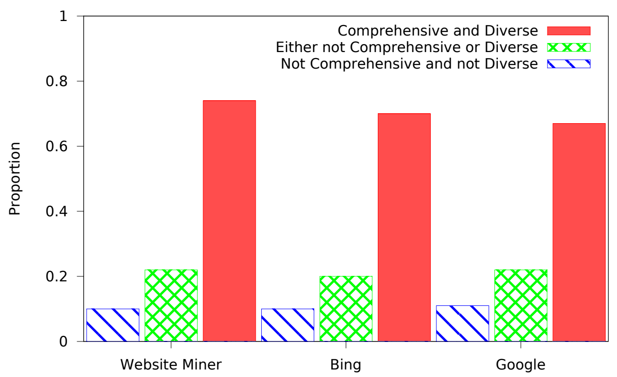


Figure 5: Proportion of query suggestion sets classified into the diversity categories for each approach.

tion that companies usually exposes their products in their websites in a hierarchical way. Products are associated to pages and relations are hyperlinks between pages. By mining these websites, our approach obtains their connectivity graph and uses mutual information to remove relevant content, and probabilistic modelling to weight the associations. An experimental evaluation in the query suggestion domain has shown that our approach produces better recommendations in terms of quality and diversity than two commercial search engines. A possible future direction for this work would be applying techniques that add semantic to the connections [19].

6. REFERENCES

[1] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 88–95. ACM, 2003.

[2] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 875–883. ACM, 2008.

[3] Z. Chen, S. Liu, L. Wenyin, G. Pu, and W. Ma. Building a web thesaurus from web link structure. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 48–55. ACM, 2003.

[4] N. Gupta. Extracting descriptions of problems with product and services from twitter data. In *Proceedings of the 3rd Workshop on Social Web Search and Mining (SWSM2011)*, 2011.

[5] J. Huang and E. Efthimiadis. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 77–86. ACM, 2009.

[6] M. Ito, K. Nakayama, T. Hara, and S. Nishio. Association thesaurus construction methods based on link co-occurrence analysis for wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 817–826. ACM, 2008.

[7] Y. Jing and W. Croft. An association thesaurus for information retrieval. In *Proceedings of RIAO*, volume 94, pages 146–160. Citeseer, 1994.

[8] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pages 387–396. ACM, 2006.

[9] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[10] R. Kraft and J. Zien. Mining anchor text for query refinement. In *Proceedings of the 13th international conference on World Wide Web*, pages 666–674. ACM, 2004.

[11] Z. Liu, S. Natarajan, and Y. Chen. Query expansion based on clustered results. *Proceedings of the VLDB Endowment*, 4(6):350–361, 2011.

[12] C. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[13] C. Manning and H. Schutze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[14] C. Matuszek, M. Witbrock, R. Kahlert, J. Cabral, D. Schneider, P. Shah, and D. Lenat. Searching for common sense: Populating CycŹ from the Web. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1430. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[16] A. Popescu. *Information extraction from unstructured web text*. PhD thesis, Univ. of Washington, 2007.

[17] Y. Qiu and H. Frei. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169. ACM, 1993.

[18] I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 213–220. ACM, 2003.

[19] R. Snow, D. Jurafsky, and A. Ng. Learning syntactic patterns for automatic hypernym discovery. *Advances in neural information processing systems*, 17:1297–1304, 2005.

[20] P. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the twelfth european conference on machine learning (ecml-2001)*, 2001.

[21] A. Yates, S. Schoenmackers, and O. Etzioni. Detecting parser errors using web-based semantic filters. *EMNLP*, 2006.