

# Harvesting Forum Pages from Seed Sites

Luciano Barbosa

Universidade Federal de Pernambuco  
Av. Jornalista Anibal Fernandes, s/n, Recife, Brazil  
luciano@cin.ufpe.br

**Abstract.** Web forums are rich sources of conversational content. Many applications, such as opinion mining and question answering, can greatly benefit from mining and exploring such useful content. A key step towards making this content more easily available is to collect conversational pages on forum sites – so-called thread pages. In this paper, we propose a two-step crawling solution for the problem of collecting thread pages in large scale. First, since thread pages are located within forum sites, we propose an inter-site crawler that locates forum sites on the Web. To do that, the inter-site crawler focuses on the Web graph neighbourhood of forum sites, and explores the content patterns of the links in this region to guide its visitation policy. Next, to collect thread pages within the discovered forum sites, we propose an intra-site crawler that finds thread pages by learning the context of links that lead to those pages and, to detect them, relies on their content and structural features. Experimental results demonstrate that both the inter-site and the intra-site crawlers are effective and obtain superior performance in comparison to their baselines.

## 1 Introduction

There is a great variety of social data available on the Web. Internet forums are social data where users hold conversations about particular subjects or topics. There are forums in diverse topics such as movies<sup>1</sup>, agriculture<sup>2</sup> and health<sup>3</sup>. Forums are also very popular: to give some numbers, as of September 2015, a big forum website – ConceptArt.org<sup>4</sup> – had more than 376 thousand users and more than 8 billion posts, another forum – Gaia Online<sup>5</sup> – had about 26 million users and 2 billion messages (source: The Biggest Boards<sup>6</sup>). This huge amount of diverse human-generated content is very helpful for a variety of applications such as opinion mining [11], question answering [17] and forum search [13].

To take advantage of such rich content, methods to collect and process forum data have been previously proposed [9, 15, 5, 2]. In this paper, we focus on the

---

<sup>1</sup> <http://www.empireonline.com/forum/>

<sup>2</sup> <http://community.agriculture.com/>

<sup>3</sup> <http://ehealthforum.com/>

<sup>4</sup> <http://www.conceptart.org/>

<sup>5</sup> <http://www.gaiaonline.com/>

<sup>6</sup> <http://www.thebiggestboards.com/>



**Fig. 1.** Overview of our two-step strategy to collect thread pages from forums.

particular problem of collecting conversational pages of forums, also known as thread pages. Previous approaches in the area of forum crawling have mainly focused on collecting thread pages within forum sites [9, 7, 16, 4]. To avoid visiting unproductive regions of those sites, they learn regular expression patterns of URLs in the navigational paths that lead to thread pages, and use these patterns to guide the crawler’s visitation policy.

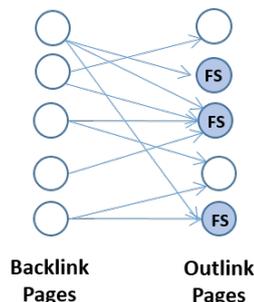
In this work, we aim to perform a broader harvest of thread pages on the Web. More specifically, we are interested not only in collecting thread pages in a particular forum site, as previous approaches [9, 7, 16, 4], but also to gather these pages from as many sites as possible. For that, we propose a two-step approach that first finds forum sites on the Web and, subsequently, collects thread pages within those sites. Figure 1 gives an overview of our solution. For the first step, we propose the Inter-Site Crawler that focuses on the Web neighbourhood of already known forum sites to discover new forum sites on the Web. Since not all links in the neighbourhood of forums lead to relevant information, we apply machine learning techniques to learn the patterns of relevant links in it. Once forum sites are found, the next step is to collect thread pages within them. This is the role of the Intra-Site Crawler. From the homepage of a given forum site, the Intra-Site Crawler navigates through the link structure of the site to locate thread pages. To focus its visitation policy on promising regions of the site, the crawler explores the context of the link neighbourhood of thread pages by using classifiers, instead of using regular expressions as previous approaches [9, 14].

The remainder of the paper is organized as follows. Section 2 describes in details the Inter-Site Crawler, and Section 3 the Intra-Site Crawler. In these sections, we also provide experimental evaluation of these solutions. Finally in Section 4, we conclude the paper.

## 2 Inter-Site Crawler

The goal of the Inter-Site Crawler is to discover forum sites on the Web. To avoid visiting unproductive regions of the Web, the Inter-Site Crawler must focus on the region of the Web where forum sites are located (site discovery) and, to collect a high-quality set of forum sites with low cost, it needs to effectively and efficiently detect forum sites (site detection). In the remaining of this section, we explain in details these two tasks.





**Fig. 3.** Example of bipartite graph used by the Inter-Site crawler to locate forum sites.

dicative words of forums such as “forum”, and “community” are visited. At the end of this process, a site is classified as a forum site if the entry-page classifier considers relevant one the visited pages of the site in the shallow crawling.

## 2.2 Site Discovery

To the best of our knowledge, no previous work has been proposed to locate forum sites on the Web. The main challenge in performing this task is that forum sites are sparsely distributed on the Web. As a result, a simple crawling strategy that randomly follows outlinks obtains a poor performance in finding forum sites, as the results in Section 2.3 suggest. To find forum sites more efficiently, we propose a crawling strategy that focuses its visitation policy on Web neighbourhood of forum sites. More specifically, the crawler explores the neighborhood graph defined by the bipartite graph composed by the backlink pages (BPs) of URLs of forum sites, and the pages pointed by BPs (outlink pages), as shown in Figure 3. The intuition behind this strategy is that a single backlink page might refer to many related pages (forum sites in our context). We call this link-rich backlink as hub page. Thus, once the crawler finds a hub page, it is only one step away from multiple forum sites. A previous work [1] used a similar strategy to locate multilingual sites on the Web.

The strategy works as presented in Algorithm 1. Initially, the user provides a set of seed URLs of forum sites. The crawler then retrieves the backlinks of these sites (line 9) using a backlink API available online<sup>7</sup>, adding them to the backlink frontier (line 10). One of the backlinks is selected from the backlink frontier (line 12) and the page that the backlink points to (backlink page) is downloaded (line 13). The outlinks of this backlink page are extracted from the page and inserted into the outlink frontier (15). Next, a link from the outlink frontier is picked (line 6) and the Site Detector verifies whether it is a forum site or not (line 8). If so, the backlinks of this link are retrieved and added to the backlink frontier, and the process continues as described before. Notice that the crawler does not

<sup>7</sup> In our experiments, we used the Mozcape API: <https://moz.com/products/api>

---

**Algorithm 1** - Site Discovery

---

```

1: Input: seeds, detector
   {seeds : URLs of forum sites, detector: Forum Site Detector.}
2: backlinkFrontier =  $\emptyset$ 
3: outlinkFrontier =  $\emptyset$ 
   {Create the empty frontiers.}
4: outlinkFrontier.addLinks(seeds)
   {Add the seeds to the outlink frontier.}
5: repeat
6:   outlink = outlinkFrontier.next()
   {Retrieve from the outlink frontier the next link to be visited.}
7:   page = download(outlink)
   {Download the content of the page.}
8:   if detector.isRelevant(page) then
9:     backlinks = collectBacklinks(page)
   {Collect the backlinks to the given page provided by a search engine API.}
10:    backlinkFrontier.addLinks(backlinks)
   {Add backlinks to the backlink frontier.}
11:   end if
12:   inlink = backlinkFrontier.next()
   {Retrieve from the backlink frontier the next link to be visited.}
13:   inpage = download(inlink)
   {Download the content of the page.}
14:   outlinks = extractOutlinks(inpage)
   {Extract the outlinks of a backlink page.}
15:   outlinkFrontier.addLinks(outlinks)
   {Add outlinks to the outlink frontier.}
16: until frontier.isEmpty()

```

---

explore the outlinks of outlink pages, and only explores backlinks of forum sites, detected by the Site Detector.

Since backlinks in the backlink frontier not necessarily lead to hub pages, and outlinks in the outlink frontier not necessarily lead to forum sites, we apply machine learning techniques to rank links in those frontiers. More specifically, the crawler uses the Backlink Classifier that predicts the likelihood of a backlink  $b$  being a hub page, given features of  $b$  such as the tokens in the URL and in the title of the page<sup>8</sup>. Likewise, the crawler uses the Outlink Classifier to predict the likelihood of a given outlink to point to a forum site based on tokens in the URL, in the anchor and around the link. The two classifiers are automatically created during the crawling process. Initially, the crawler starts with no link prioritization. After a specified number of crawled pages, a learning iteration is performed by collecting the link neighbourhood of the links that point to relevant and non-relevant pages in each set. The result of this process is used as training data for the Backlink and Outlink classifiers.

---

<sup>8</sup> The title of the page is usually provided by the backlink apis.

Technique	Prec	Rec	F-1	Acc
Naive Bayes	0.705	0.925	0.8	76.8
Logistic Regression	0.781	0.851	0.814	80.5
SVM	0.855	0.791	0.822	82.8

**Table 1.** Results from different machine learning algorithms.

Min. Likelihood	Prec	Rec	F-1
0.5	0.855	0.791	0.822
0.6	0.91	0.77	0.83
0.7	0.94	0.76	0.84
0.8	0.96	0.74	0.83
0.9	0.96	0.73	0.83

**Table 2.** Results of varying the minimum likelihood of a page being considered relevant by the entry page classifier.

### 2.3 Experimental Evaluation

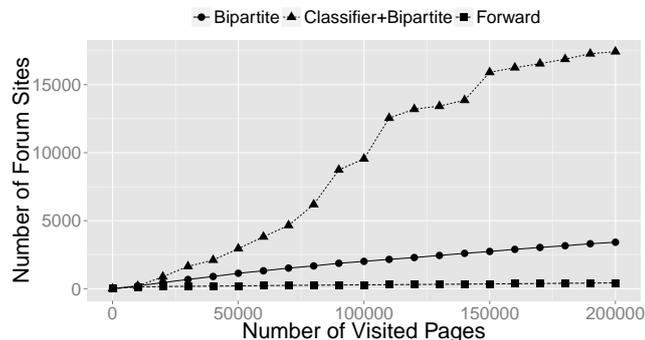
In this section, we evaluated the two tasks performed by the Inter-Site Crawler – site detection and discovery – presented previously.

**Site Detection.** To measure the quality of the Site Detector, we manually labeled 380 Web pages (182 positive and 200 negative) from a variety of sites on the Web. Positive examples are entry pages to forums in the sites. We used two thirds of the data for training and one third for test.

Table 1 presents the precision, recall, F-1 and accuracy values for 3 different machine learning algorithms. The classifiers were created using the Weka package [8] with default values. The numbers show that Support Vector Machine (SVM) obtained the best results (precision=0.855, recall=0.791, F-1=0.822, accuracy=82.8). The SVM version used was the probabilistic SVM [12], since we are interested in the class likelihood of the instances. A threshold over the class likelihood can be used, for instance, as a filter to improve the precision of the Site Detector. This filter is very useful in this environment whereby the proportion of negative examples is much higher than the positive ones. For this purpose, we varied the minimum likelihood for a page be considered relevant from 0.5 to 0.9, as presented in Table 2. For each value, we measured its quality (precision, recall and F-1). As expected, the minimum likelihood is directly proportional to the precision and inversely proportional to the recall. For instance, when the minimum likelihood is 0.9, i.e., only pages with likelihood higher 0.9 are considered relevant by the detector, the precision is 0.96.

**Site Discovery.** To evaluate our strategy to locate forum sites, we executed the following crawling configurations:

- Forward Crawler (Forward): The forward crawler randomly follows the forward links. Only out-of-site links are considered, i.e., it excludes from the crawling links to internal pages of the sites;



**Fig. 4.** Performance of the 3 strategies of inter-site crawling.

- Bipartite-Graph Crawler (Bipartite): our strategy focusing on the bipartite graph composed by backlink and outlink pages without any prioritization of links in the graph;
- Classifier-Based Bipartite-Graph Crawler (Bipartite+ Classifiers): our strategy using classifiers to prioritize links in the bipartite graph.

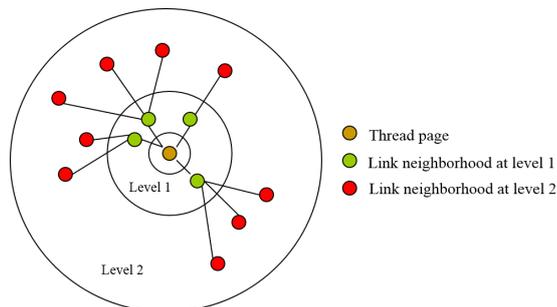
Each configuration visited 200,000 pages. Only 2 forum sites were provided as seeds to start the crawl<sup>9</sup>. The performance of the crawling strategies was measured by the total number of forum sites collected. The minimum likelihood used by the Site Detector to consider a site as relevant was 0.9, since we are interested in obtaining a high-quality collection of forum sites.

Figure 4 presents the number of collected forums versus the number of visited pages for each approach. At the end of the crawling processes, the Bipartite+Classifiers approach collected the highest number of forum sites (17,429 sites). That is, only from 2 seed URLs, our best strategy discovered more than 17K forum sites. The Bipartite crawler located fewer sites (34,515) followed by the Forward crawler (427 sites). These results show that (1) the bipartite graph strategy is in fact effective: the Bipartite crawler found 8 times more sites than the baseline (the Forward crawler); and (2) the classifiers (Backlink and Forward classifiers) used to prioritize the links in the bipartite graph hugely improved the performance of the bipartite crawler: Bipartite+Classifiers crawler discovered 5 times more for forum sites than the Bipartite crawler.

### 3 Intra-Site Crawler

The pages in forum sites that contain users’ conversations are called thread pages. The goal of the Intra-Site Crawler is to locate thread pages in a given forum site. To achieve that, it needs to locate and detect thread pages in forum

<sup>9</sup> Seeds: <http://www.woodworkforums.com/> and <http://ubuntuforums.org/index.php>



**Fig. 5.** Example of context graph for thread pages (adapted from [6]).

sites. For the first task, it is important that the crawler avoids unproductive regions of the sites that might not lead to thread pages. For the second one, the crawler needs to perform a high-quality detection, otherwise the repository of thread pages collected by the crawler might have poor quality. We give further details about these tasks in the remaining of this section.

### 3.1 Locating Thread Pages

Thread pages are only a subset of the pages in forum sites. In addition to them, forum sites might contain pages related to documentation, news etc. The Intra-Site crawler then needs to focus on the region of the website where thread pages are located to collect the maximum number of thread pages visiting the lowest number of pages as possible. For that, the crawler explores the patterns of links inside forum websites using context graphs [6]. Figure 5 presents an example of a context graph of two levels. The thread page is located in the center of the context graph. The main assumption of context graphs is that links at the same distance to the thread page in this graph have similar context. For instance, in the site `ubuntuforums.org`, URLs that point to thread pages (one step away) have the string “`showthread`” in common, whereas URLs two steps away have the string “`forumdisplay`” in common. Context graphs have been used to locate pages in other contexts such as pages in a given topic [6] or pages containing Web forms [3]. The Link Classifier is the component in the intra-site crawler that leverages context graphs to locate thread pages. More specifically, the Link Classifier estimates the distance (number of edges) of a given link to a thread page based on its context. The context is composed by the tokens in the URL, anchor and words around the anchor, which are the features used by the classifier.

The Link Classifier is automatically built as the crawl progresses. Initially, the crawler starts with no link prioritization. After a specified number of crawled pages, the links visited that led to thread pages collected so far, which compose the context graph, are used as training data to build the Link Classifier.

The screenshot shows a forum thread with three posts. Each post includes a user profile picture, name, location, post count, review count, and a 'Reply' button. The first post is by 'williams3305' from Liverpool, dated Apr 11, 2014, 1:02 PM. The second post is by 'TokoTheTraveler' from Finland, dated Apr 11, 2014, 2:37 PM. The third post is by 'joffheadat' from Manila, dated Sep 30, 2014, 5:15 AM.

(a) Thread page

1-20 of 956 topics

« 1 2 3 4 5 6 7 8 9 ... 48 »

Forum	Topic	Replies	Last post
Pas de la Casa	<a href="#">pas de la casa snow 2014/2015</a> by Blair T	1	yesterday by StookportBre...
Andorra	<a href="#">ski conditions in late feb?</a> by dannycrow85	2	yesterday by StookportBre...
Andorra	<a href="#">Ski lifts open in the summer?</a> by Geoff B	7	yesterday by Geoff B
Andorra	<a href="#">Andorra/Pyrenees in April</a> by kpter	11	yesterday by Luis M
Pas de la Casa	<a href="#">Cost this year</a> by Jonny6811	0	Dec 07, 2014 by Jonny6811
Soldeu	<a href="#">When does the ski season start?</a> by petermoore	7	Dec 07, 2014 by petermoore

(b) Index page

Fig. 6. Examples of an index and a thread page.

### 3.2 Detecting Thread Pages

In order to collect a high-quality set of thread pages from a given site, the intra-site crawler needs to perform an effective thread page detection. Thread pages are composed of user posts (see Figure 6(a)). Posts are usually within records, which contain, in addition to posts, meta-information about the posts such as the user who posted the information, the date of posting etc. In a previous work [2], we implemented a method to extract records from forum pages. Records also appear in other types of pages in forum sites such as the ones that point to thread pages (a.k.a. index pages). Figure 6(b) presents an example of index page.

Based on these observations, our thread page detection relies on the record extraction to obtain features for the classification. Only pages with extracted records are considered for classification. Thus, given a forum page, the first step of the detection is to extract records from the page. If records are extracted,

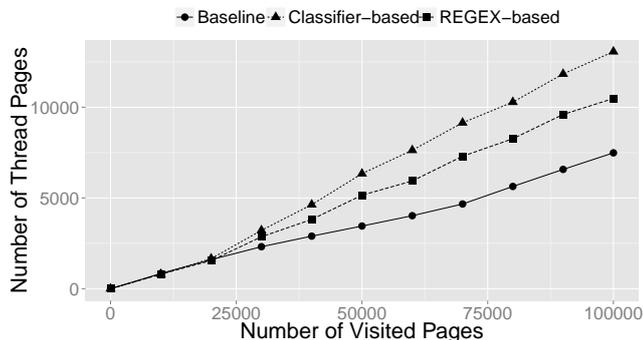


Fig. 7. Performance of the 3 strategies of intra-site crawling.

thread classification is applied over the records to verify whether the page is a thread page or not. Records of index pages have different layouts from records of thread pages as one can see in Figure 6(a). For instance, records in thread pages usually have longer texts than in index pages; and records in index pages contain internal links to thread pages which is not always the case for records from thread pages. Based on that and similar to [9], we employ the following set of features based on the layout of the records:

- Average number of date and user information for all records in the page. To detect information about date and user in a record, we used detectors based on regular expressions [2];
- Average size, variance and noise to signal (standard deviation/average) of the texts in the records;
- Average number of images, internal and external links.

### 3.3 Experimental Evaluation

In this section, we evaluate the thread-page detection and discovery presented previously.

**Thread-Page Detection.** To build the training data, we labeled 1280 instances (50% positive and 50% negative) and split in 722 for training and 558 for test. We tried different machine learning techniques available in Weka package [8] with their default values. The multi-layer perceptron obtained the best results: Accuracy = 92.2%, Precision = 0.91, Recall = 0.87 and F-measure = 0.89.

**Thread-Page Discovery.** For the problem of thread-page discovery, we implemented 3 different strategies of prioritization of links:

- Baseline: the baseline randomly follows the internal outlinks of the pages;
- REGEX-based: the regex-based strategy follows links using regular expressions learned from URLs of thread pages, as proposed by [10] and used by Jiang et al. [9] in their forum crawler;

- Classifier-based: the classifier-based approach is the one proposed in this paper for the Intra-Site crawler, which builds a Link Classifier learned from the context graph as we described previously. Probabilistic SVM [12] was the learning algorithm used by the Link Classifier, and it was created with default values for SVM provided in Weka [8].

The 3 approaches collected thread pages within 586 forum sites, and each one visited a total of 100,000 pages of these sites. We evaluated the approaches based on the number of thread pages collected by each approach, identified by the thread page detector. The minimum likelihood used by the thread page detector to consider a page as a thread page was 0.8. Both the REGEX-based and the Classifier-based crawlers start with the same link prioritization than the baseline. The learning process to create the regular expressions, for the REGEX-based crawler, and the link classifier, for the Classifier-based crawler, was performed every 20,000 visited pages.

Figure 7 presents the results for the 3 strategies. As expected, until 20,000 visited pages all approaches had similar behaviour. After that, the approaches presented different performance since the Classifier-based and REGEX-based approaches started running the learning process. At the end of the crawling process, our approach outperformed the other two: the Classifier-based crawler collected 13,063 thread pages, whereas the REGEX-based collected 10,483 and the Baseline 7,486. A possible reason why our strategy obtained better results than the REGEX-Based one is that the REGEX-Based crawler learns patterns only from the URLs of relevant pages, whereas our strategy leverages patterns using machine learning not only from URLs but also from the anchor of these links and the context around them.

## 4 Conclusion

We presented in this paper a two-step crawling approach to collect conversational pages on the Web. First, the Inter-Site Crawler discovers forum sites from seed sites. For that, it restricts its crawl to the link neighbourhood composed of the backlink pages of forum sites, and the outlink pages of backlink pages. To prioritize links in this graph, we apply machine learning techniques. The URLs of the discovered forum sites are provided to the Inter-Site Crawler that collects the conversational pages in these sites. To more efficiently locate those pages, it uses a link classifier that explores the context of the links around conversational pages. Our experimental evaluation shows that our crawling approaches harvested high-quality collections and are more efficient than the baselines. As future work, we plan to apply this strategy to specific domains, instead of using it in a generic manner as we presented in this paper.

## References

1. L. Barbosa, S. Bangalore, and V. K. R. Sridhar. Crawling back and forth: Using back and out links to locate bilingual sites. In *Proceedings of the 5th International*

- Joint Conference on Natural Language Processing*, pages 429–437, 2011.
2. L. Barbosa and G. Ferreira. Extracting records and posts from forum pages with limited supervision. In *International Conference on Web Information Systems Engineering*, pages 233–240. Springer, 2015.
  3. L. Barbosa and J. Freire. Searching for hidden-web databases. In *WebDB*, pages 1–6, 2005.
  4. R. Cai, J.-M. Yang, W. Lai, Y. Wang, and L. Zhang. irobot: An intelligent crawler for web forums. In *Proceedings of the 17th international conference on World Wide Web*, pages 447–456. ACM, 2008.
  5. G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474. ACM, 2008.
  6. M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, M. Gori, et al. Focused crawling using context graphs. In *VLDB*, pages 527–534, 2000.
  7. Y. Guo, K. Li, K. Zhang, and G. Zhang. Board forum crawling: a web crawling method for web forum. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 745–748. IEEE Computer Society, 2006.
  8. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
  9. J. Jiang, X. Song, N. Yu, and C.-Y. Lin. Focus: learning to crawl web forums. *Knowledge and Data Engineering, IEEE Transactions on*, 25(6):1293–1306, 2013.
  10. H. S. Koppula, K. P. Leela, A. Agarwal, K. P. Chitrapura, S. Garg, and A. Sasturkar. Learning url patterns for webpage de-duplication. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 381–390. ACM, 2010.
  11. B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.
  12. J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.
  13. J. Seo, W. B. Croft, and D. A. Smith. Online community search using thread structure. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1907–1910. ACM, 2009.
  14. M. L. Vidal, A. S. da Silva, E. S. de Moura, and J. Cavalcanti. Structure-driven crawler generation by example. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 292–299. ACM, 2006.
  15. H. Wang, C. Wang, C. Zhai, and J. Han. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 435–444. ACM, 2011.
  16. Y. Wang, J.-M. Yang, W. Lai, R. Cai, L. Zhang, and W.-Y. Ma. Exploring traversal strategy for web forum crawling. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 459–466. ACM, 2008.
  17. B. Webber and N. Webb. Question answering. *The handbook of computational linguistics and natural language processing*, pages 630–654, 2010.