Seed Selection for Domain-Specific Search

Pattisapu Nikhil Priyatam Ajay Dubey Krish Perumal

Sai Praneeth Dharmesh Kakadia Vasudeva Varma

Search and Information Extraction Lab
IIIT-Hyderabad
AP, India
{nikhil.priyatam, ajay.dubey, krish.perumal, dharmesh.kakadia}@research.iiit.ac.in,
saipraneeeth@gmail.com, vv@iiit.ac.in

ABSTRACT

The last two decades have witnessed an exponential rise in web content from a plethora of domains, which has necessitated the use of domain-specific search engines. Diversity of crawled content is one of the crucial aspects of a domain-specific search engine. To a large extent, diversity is governed by the initial set of seed URLs. Most of the existing approaches rely on manual effort for seed selection. In this work we automate this process using URLs posted on Twitter. We propose an algorithm to get a set of diverse seed URLs from a Twitter URL graph. We compare the performance of our approach against the baseline zero similarity seed selection method and find that our approach beats the baseline by a significant margin.

1. INTRODUCTION

The ever increasing data on WWW poses unprecedented challenges for search engines to come up with relevant results. This is due to the "one size fits all" strategy adopted by search engines. On the contrary, domain-specific search engines cater to a specific audience (or specific needs of a general audience) while offering high quality search within a particular domain. CiteSeerX is one such search engine that offers specialized search for computer science articles. Few other search engines are *Coremine* [1] and *Flipdog* [2] which offer search for medical domain and job listings respectively [5].

A domain-specific search engine cannot and will not be used if the search engine does not contain crawled content of its entire domain. Despite restricting itself to one domain, if it is not able to serve every query in that domain, it is a useless task to build one.. Hence crawl diversity is one of the crucial factors that impact the quality of domain-specific search. To a large extent, crawl diversity depends on the choice of seed URLs (the list of URLs that the crawler starts with). Though significant effort has gone into building various crawling strategies, not enough research has been done in choosing good quality seed URLs.

Generally, seed URLs are collected by manually framing domain-specific queries, firing them on to a search engine and filtering the retrieved results. While collecting these URLs the following questions need to be answered: Is the seed URL set representative of the entire domain? Is it sufficiently diverse to contain URLs from various sub-topics / sub-domains of the domain? What should be the size of the URL list? Answering all these questions requires domain expertise and manual effort. To the best of our knowledge, no automated system exists for this purpose. In this work, we present an approach to automate the process of seed URLs collection for domain-specific search with a special focus on "diversity". We do not attempt to solve the problem of estimating the number of seed URLs per domain.

Any automated approach for seed URLs collection requires a huge collection of candidate URLs to start with. Such a collection should contain URLs coming from diverse sources like blogs, organizational websites, etc. Moreover, there should exist some way of quantifying the similarity between URLs without having to explicitly crawl them. An ideal collection of such URLs should pertain to content coming from all domains and in the same proportion as that of WWW.

Given the proliferation of Web 2.0 and social media, user generated content is becoming a significant part of WWW. Social media draws participation of people from various social and geographical backgrounds. It contains content posted by experts and enthusiasts of a variety of domains. A significant proportion of content consists of URLs. Additionally, it contains information about the context in which the URLs were posted by the user at a particular point of time. Due to these reasons, social media acts as an ideal source of seed URLs. In this work, we use Twitter data for automatically collecting diverse seed URLs. The diversity of a seed URL set is measured by the diversity of the resulting web crawl.

2. RELATED WORK

The problem of seed URL selection has not received sufficient attention in the past. But recently, Zheng et al. [13]

[1] http://www.coremine.com
[2] http://www.flipdog.com

 WWW'14 Companion, April 7-11, 2014, Seoul, Korea. ACM 978-1-4503-2745-9/14/04. http://dx.doi.org/10.1145/2567948.2579216

presented a graph based approach to select seed URLs for web crawlers. For this selection, they employ several seed selection strategies based on PageRank, number of outlinks and website importance. They compare the performance improvements of their approach over random seed selection (baseline). It is worth noting that this work analyzes the quality of a seed by crawling the corresponding web page and analyzing its page content. Hence, in a scenario where there are millions of candidate URLs, this approach proves to be computationally expensive. Also, this work does not address the problem of domain-specific seed URL selection. Dmitriev [3] proposes a host-based seed selection method. They use measures of quality, importance and potential yield of hosts for selecting a document of the host as a seed. This selection is carried out based on the geographic region to which the host belongs; the host-trust score that gives indication of popularity, trustworthiness, reliability and quality; the number of links in a document pointing to other documents within that host; the probability of spam in the host; and the expected yield of hosts (calculated using past crawl statistics). Prasath et al. [10] use a manually assigned relevance score and a gain-share score (proposed by them) to decide on potential seed URLs. However, this work does not report the contribution of each of the scores individually. Hence, its contribution is unknown and limited.

The rising popularity of Twitter in recent years has prompted researchers to use it to solve a variety of problems in IR. Shankar et al. [11] define different ways of tapping into the collaborative wisdom of the crowd using Twitter.

Leveraging the fact that many tweets refer to and contain named entities, Finin et al. [6] use crowdsourcing platforms like MTurk and CrowdFlower to annotate named entities in Twitter. Phelan et al. [9] show that mining tweets can provide access to emerging topics and breaking events. They present a novel news recommendation system, called Buzzer, that harnesses real-time Twitter data as the basis for ranking and recommending articles from a collection of RSS feeds. Yan et al. [12] present a co-ranking framework for a tweet recommendation system that takes popularity, personalization and diversity into account.

Castillo et al. [2] assess the credibility of information from tweets. They report the use of a number of features like the time for which a user has been on Twitter, number of times a user has tweeted, number of retweets, number of followers and whether a tweet contains URLs or not. Among other things, they conclude that credible news are propagated through authors who have previously written a large number of messages, originate at a single or a few users in the network, and have many retweets.

Mishne et al. [8] use tweets and tweet conversations as anchor text to enrich document representations. Tweets referencing web pages provide a valuable source of content not found in the pages themselves. One can get information about breaking news in real time from tweet text since they contain relevance signals that are unlikely to be found anywhere else at the time the tweet was posted.

Dong et al. [4] aim to tackle the problem of realtime web search using Twitter. Realtime web search involves quickly crawling relevant content and ranking documents with impoverished link and click information. They use the Twitter data stream to detect fresh URLs, and also to compute novel and effective features for ranking these URLs. However, they have not tackled the problem of URL diversity, which is vital for quality web search. Boanjak et al. [1] present a crawler which allows retrieval of Twitter data from a focused community of interest. Since the proposed system is modular, one can focus it on different segments of Twitter data, namely different communities of users described by geographic, demographic, linguistic or even topical characteristics.

Menczer et al. [7] propose various methods for evaluating topic specific crawl. In the *Assessment via Classifiers* method, they train a classifier for each topic and evaluate the precision of the crawled set. This requires huge amount of accurate training data (manual tagging) which is labour intensive. The second method *assessment via a retrieval system* is based on the intuition that a crawler should retrieve good pages earlier than the bad ones. The last method *Mean Topic Similarity* measures the cohesiveness of the crawled set with the topic as the core. The underlying assumption is that the more cohesive the crawled set the more relevant its pages. To the best of our knowledge no method exists which measures diversity within a crawl.

3. PROPOSED SYSTEM

As mentioned in the introduction section, we use URLs posted on Twitter as candidates for selecting seed URLs. Twitter is an online social networking and micro-blogging service that enables its users to send and read text-based messages of up to 140 characters, known as "tweets" [3]. Twitter provides the following information about its users: Twitter id, name, location, user description, URL of the user, followers, etc. Additionally, tweets contain hashtags, retweets, mentions and URLs. We use Twitter data for the automatic collection of seed URLs because of the following reasons.

- About 25% tweets contain URLs [4], many of which are pointers to information rich portals.
- Users post and exchange information about a variety of trending topics like entertainment, politics, tourism, etc.
- Twitter has millions of users coming from different social, geographical and cultural backgrounds which ensures a diverse audience.
- Twitter provides to the users the option of following other users. Moreover, a user's tweet can be endorsed by other users in the form of retweets(RT). Using the follower-followee relationship and retweet (RT) information, we can model the similarity between users.
- The diversity of user opinions can be measured using content overlap between tweets.
- The trail of tweets over time can be used to ensure temporal diversity.
- Huge number of new URLs are posted everyday. Following these URLs would lead to a fresh and updated crawl.

Figure 3 demonstrates the architecture of our system. The individual components of the system are briefly described here:

[3]http://en.wikipedia.org/wiki/Twitter
[4]http://techcrunch.com/2010/09/14/twitter-event/

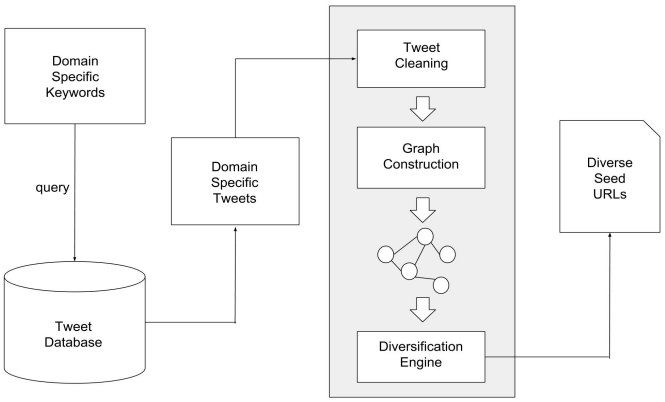

Figure 1: System Architecture

Domain specific keywords: These are the words specific to the domain. These are manually fed into the system. These typically consist of name of the domain (entertainment, tourism, etc) and its synonyms. A good set of keywords is one, which when queried on any corpus, can retrieve almost all documents belonging to that domain.

Twitter Search: We query a Twitter corpus available on a local disk with the keywords obtained above to get domain-specific tweets.

Cleaning tweets: This further refines the above search to output only those tweets that contain a URL. Also, it eliminates all the expired and invalid URLs. It also gets the original redirected URLs from URL shortening services like *bitly, tinyurl*, etc. We refer to these tweets as cleaned tweets.

Graph construction: This module constructs an undirected unweighted graph from the cleaned tweets. Each URL in the graph is a node/vertex and two vertices share an edge if they are found to be similar.

Diversification Engine: The graph thus generated is fed to the diversification engine which returns 'k' diverse URLs.

4. PROPOSED APPROACH

We pose the problem of diverse seed selection as a graph search problem. As mentioned in section 3 we form a graph where each vertex is a URL shared on Twitter. Two vertices are connected if they are "similar". Algorithm 1 shows how the diversification of seed URLs is done. We propose various methods of computing "similarity" between these URLs which are explained in the next section.

Figure 2 shows the working of the algorithm with an example. In this example we have nine vertices and we need the three most diverse seed URLs (k=3). As we can see that the final set of URLs returned by the algorithm are far apart from each other and hence are diverse.

Having explained our diversification algorithm, we now propose different ways of computing similarity between Twitter URLs. Later, we compare the performance of each of these against a basic approach called Zero Similarity.

Algorithm 1 Diversification Algorithm

Input : *Graph* $G(V, E)$, *number of seeds* k, $k < |V|$
Output : *Diverse k seed URLs*
Initialize Picked Nodes $P = \{\varnothing\}$, *Eliminated Nodes* $E_l = \{\varnothing\}$, $hops = |V| - 1$
while $|P| <= k$ **do**
  Pick random node n such that $n \in V$ *and* $n \notin P$ *and* $n \notin E_l$
  Add n to P
  Add neighbours(n, h) *to* E_l
  if $|E_l \cup P| = |V|$ **then**
    Reinitialize $P = \{\varnothing\}$, *Eliminated Nodes* $E_l = \{\varnothing\}$
    $h = h - 1$
  end if
end while
Return P

5. GRAPH CONSTRUCTION

In the previous section we have seen how to extract the most diverse 'k' URLs given a graph of connected URLs. Thus far, we have assumed that we already have a graph where similar URLs are connected. This section explains various ways in which such a graph can be constructed using cleaned tweets.

5.1 Content Similarity

We define two URLs to be similar if the tweets that contain these URLs have content overlap above a threshold. This is based on the intuition that tweets with significant content overlap are talking about the same thing, and the URLs present in the tweets may be pointing to similar sort of resources/information and hence these URLs are not diverse. Equation 1 shows the formula to calculate content similarity, where T_i and T_j are the sets of words in tweets i and j respectively. Equation 1 is nothing but Jaccard Index, which in this case is used to calculate tweet content

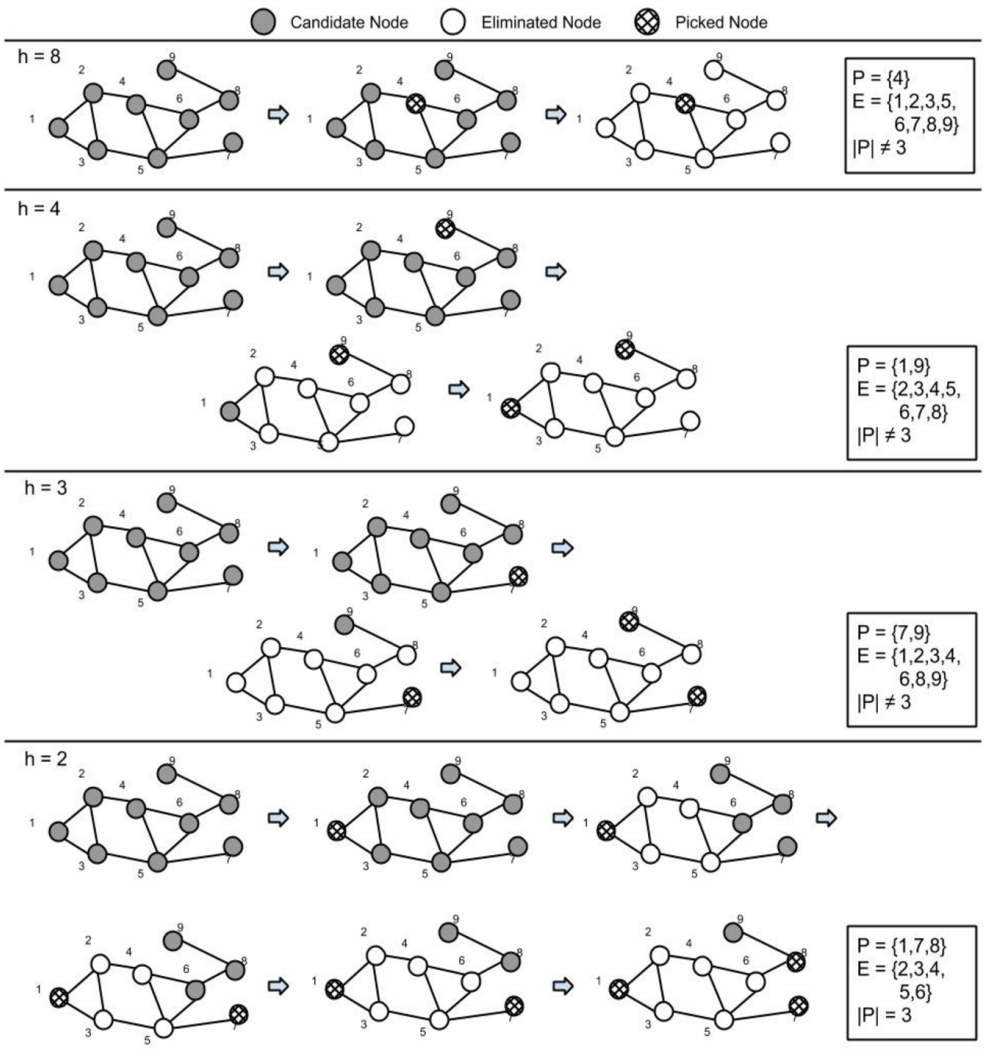

Figure 2: Example of algorithm 1, number of seeds k=3

similarity.

$$Content\ Similarity\ = \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \quad (1)$$

5.2 URL N-Grams Similarity

We define two URLs to be similar if the URL n-grams have overlap above a threshold. This is one of the most intuitive measures since URLs with significant N-gram overlap point to similar sort of information. Since most of the times URLs contain short forms, noise and spell variations, we do not use URL token overlap. Instead, considering n-grams handles most of these problems to a sufficient degree of satisfaction. In this work we experiment with 4-grams. The process of overlap calculation is shown below:

URL: http://www.flickr.com/photos/tedfriedman/36855/
Cleaned URL Content: flickrphotostedfriedman
4-Grams: {flic, lick, ickr, ckrp, krph, rpho, ...}

$$URL\ Similarity\ = \frac{|U_i \cap U_j|}{|U_i \cup U_j|} \quad (2)$$

Equation 2 shows the formula to calculate URL n-grams similarity, where U_i and U_j are the sets containing 4-grams of URLs i and j respectively.

5.3 User Similarity

In this approach we use a simple notion, *birds of the same feather flock together*. Two users are considered similar if at least one of them has retweeted the other's tweet. URLs posted by similar users are also considered similar. This is based on the assumption that, if one user retweets another user's tweet, both users share common thinking/knowledge and are likely to post similar content/URLs. The same can be done in a much better way using the follower-followee relationship in Twitter. However, in absence of this information, we have used information from retweets. This approach is meant to capture user diversity.

5.4 Zero Similarity

In this graph, no two URLs share an edge, i.e. no two URLs are considered similar and hence the name "Zero Similarity". If such a graph is given as an input to Algorithm 1, it would merely return any random 'k' URLs from the graph, this is our baseline system.

It would be noteworthy to mention that the timestamp of a tweet can also be used to construct a graph. The timestamp of a tweet consists of the date and time at which it was tweeted. We can define two URLs to be similar if the tweets containing these URLs were posted within a predefined time interval. The rationale behind this approach is that tweets posted at around the same time might talk about the same event/entity. The purpose here is to get a temporally diverse set of URLs. We do not use this method to construct the graph since we are working with a dataset of tweets of 30 days.

6. EVALUATION METRIC

The diversity of a seed URL set is judged by the diversity of the web crawl that it leads to. So to measure each Seed set (obtained by various graphs), we explicitly crawl each one of those sets. To measure the crawl diversity we use *dispersion*. Dispersion refers to the spread or variability in a variable. Variance, standard deviation and interquartile range are widely used measures of statistical dispersion. We measure the variance across the crawled set of documents to judge its diversity. The dispersion, as shown in equation 3, is calculated as the average squared distance of all documents from the mean. Here $\vec{d_i}$ refers to document i represented as a bag of words vector, $\vec{\mu}$ represents the mean of all $\vec{d_i}\,'s$ and N represents total number of documents selected.

$$D2\ Score\ = \frac{\sum_{i=1}^{N}(\vec{d_i} - \vec{\mu})^2}{N} \tag{3}$$

One problem with using dispersion as a metric to judge diversity is that, irrelevance can often be misunderstood as diversity. A typical domain-specific crawl often contains out-of-domain pages, ill-parsed pages, advertisement pages etc. All these are irrelevant to us and all such irrelevant content would add up to account for a huge dispersion score, which is undesirable. To avoid this, we do not compute dispersion over entire crawl, we compute it over relevant documents of the crawl. Relevant documents are picked from the crawl manually. To avoid parsing errors, the content of the web pages are manually copied into text files (We do not use a crawler)

7. EXPERIMENTAL SETUP

We conduct our experiments using a dataset of fifty million tweets collected over a period of one month (Nov 2009). Out of that we filter tweets coming from **tourism** domain (about 7500 in number, these can be downloaded at [5] for further inspection). These are the domain-specific tweets that we work on. On an average the graph construction takes roughly 45 minutes on a single machine of *2 GB RAM* and an *Intel Core 2 duo processor (2.53 GHz)*. Once the graph is constructed, the diversification algorithm returns the diverse seed URLs within 5 seconds.

[5] `https://docs.google.com/file/d/0B_9ISEpIrWxEd0ljaEplUDhEZWc/edit?usp=sharing`

8. RESULTS

The cumulative performance of each approach is shown in table 1 where 'ZS' indicates zero similarity and 'Cont' indicates content. In content and URL based similarity metrics, we have a notion of threshold i.e. we define two URLs to be similar if their content / URL token overlap is above a threshold. Table 1 shows the performance of content Content, URL based similarity measures across various thresholds ranging from 0.15 to 0.35. Note that, a threshold value of 0 would mean all URLs in the graph are connected and a threshold value of 1 would mean no two URLs in the graph are connected.

Threshold	0.15	0.20	0.25	0.3	0.35
ZS	35.0	35.0	35.0	35.0	35.0
Cont	69.1	36.1	31.7	36.4	38.4
URL	34.2	39.9	42.7	33.9	39.8
User	32.0	32.0	32.0	32.0	32.0
Cont + URL	49.9	47.6	34.7	59.5	42.6
User + URL	44.4	29.2	38.8	35.4	35.6
User + Cont	64.2	29.4	36.6	42.0	32.7
Cont + URL + User	**62.6**	**63.4**	51.4	32.6	29.6

Table 1: Dispersion

9. ANALYSIS

We can clearly see from table 1 that the Content + URL + User approach of graph construction outperforms the rest. We also observe that, while the Content, URL and user do not show clear supremacy over zero similarity approach (baseline), their combinations (Content + URL, User + URL and Content + User) easily beat the baseline. The performance of all approaches becomes randomized when the threshold reaches 0.3. This is due to the hard constraint that tweets must satisfy, i.e. a minimum 30 percent of tweet content or URL overlap must be there so that 2 URLs can be connected. In such a scenario, the URL graph is sparsely connected and hence the diversification algorithm ends up picking most of the URLs in a randomized fashion.

10. DRAWBACKS

Though we are successful in achieving diversity within our seed URL set, our work suffers from several drawbacks:

- Our evaluation involves manual labour and is often costly, this allowed us only depth 1 crawl.
- Our approach heavily depends on data source (Twitter in this case), this does not allow us to extend our approach for building domain-specific search engines for other languages. Also it restricts us to only those domains whose URLs can be easily found on social media, unlike some less popular domain like "Nuclear Physics"
- The graph construction is time consuming (order of V^2 computations)

11. CONCLUSIONS AND FUTURE WORK

In this work, we have made several contributions to the field of domain-specific crawling. We define the problem of seed selection for domain-specific search and propose an automated solution for it. This is the first attempt to tap social media for solving the problem of seed selection. Moreover, this work gives special treatment to the problem of capturing diversity in domain-specific crawling. Though we have been focusing on domain-specific search throughout this work, diversity plays a key role in generic search engines as well. Our experiments reveal that the combination of Content, URL and user approach outperforms the zero similarity approach. This shows significant evidence that some kind of similarity between URLs can indeed be captured using social media. This work has the potential to usher in new fields in crawling. The final conclusion that we draw is that this new field of research has a lot of promise and will become quintessential for building search engines in near future.

The work presented in this paper has a lot of scope for future enhancements. In step 2 of algorithm 1, the process of random selection gives each URL an equal probability of getting picked. This step can be tweaked to get the most relevant URL, provided we have a metric to measure relevance of URLs to the domain. In our approach, factors like location of the user can be incorporated to introduce geographical diversity. One could also score the URLs based on how good they are as seeds by using measures like number of outlinks. With respect to experimentation, state-of-the-art in query expansion can be used to get tweets that are much more representative of the domain. Apart from Twitter, URLs can also be mined from other social media networks like Facebook. We would like to measure the quality of crawl for domains which have less presence on social media like thermodynamics and nuclear physics.

We are evaluating the diversity of a seed set using the diversity of the resultant crawl. There is one interesting thing to note here. Suppose we give seeds S1 and S2 to two instances of a crawler program, stop them after five days, and evaluate diversity. There's no saying the order won't be flipped if we let both run for one more day. To tackle this issue we would like to study diversity measure of a seed set at different time slices of the crawl. We would like to compare our seed URLs with existing sets of domain-specific URLs like the ones present in *Open Directory Project* [6]. We also aim to compare against manual seed selection using crowdsourcing platforms like *Amazon Mechanical Turk* [7] and *CrowdFlower* [8].

Thus far, our discussion was restricted to the use of unsupervised evaluation techniques like dispersion. We wish to use supervised evaluation techniques when we have a predefined subtopic structure for a domain. Consider the example of the tourism domain with the following subtopic structure: information about religious places, historical places, tourist spots, travel services, famous cuisines, best season to visit, weather reports, shopping centres, etc. We could use this subtopic structure (domain knowledge) to evaluate diversity in our crawl. This would give us a definite and a clearer picture of diversity "specific to our domain".

[6]http://www.dmoz.org/
[7]http://www.mturk.com
[8]http://crowdflower.com/

12. REFERENCES

[1] M. Boanjak, E. Oliveira, J. Martins, E. Mendes Rodrigues, and L. Sarmento. Twitterecho: a distributed focused crawler to support open research with twitter data. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 1233–1240. ACM, 2012.

[2] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM, 2011.

[3] P. Dmitriev. Host-based seed selection algorithm for web crawlers, 2008. US Patent App. 12/259,164.

[4] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th international conference on World wide web*, pages 331–340. ACM, 2010.

[5] D. Fesenmaier, H. Werthner, and K. Wober. Domain specific search engines. In *Travel Destination Recommendation Systems: Behavioural Foundations and Applications*, pages 205–211. CABI, 2006.

[6] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88. Association for Computational Linguistics, 2010.

[7] F. Menczer, G. Pant, P. Srinivasan, and M. Ruiz. Evaluating topic-driven web crawlers. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 241–249. ACM, 2001.

[8] G. Mishne and J. Lin. Twanchor text: a preliminary study of the value of tweets as anchor text. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1159–1160. ACM, 2012.

[9] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388. ACM, 2009.

[10] R. Prasath and P. Öztürk. Finding potential seeds through rank aggregation of web searches. In *Proceedings of the 4th international conference on Pattern Recognition and Machine Intelligence*, pages 227–234. Springer, 2011.

[11] K. Shankar and M. Levy. *# Crowdsourcing Tweet Book01: 140 Bite-Sized Ideas to Leverage the Wisdom of the Crowd*. Thinkaha, 2011.

[12] R. Yan, M. Lapata, and X. Li. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual meeting of the Association for Computational Linguistics*, pages 516–525. ACL, 2012.

[13] S. Zheng, P. Dmitriev, and C. Giles. Graph based crawler seed selection. In *Proceedings of the 18th international conference on World wide web*, pages 1089–1090. ACM, 2009.