



Title	Towards a Novel and Timely Search and Discovery System Using the Real-Time Social Web
Authors(s)	Phelan, Owen, McCarthy, Kevin, Smyth, Barry
Publication date	2013-04-06
Publication information	Phelan, Owen, Kevin McCarthy, and Barry Smyth. "Towards a Novel and Timely Search and Discovery System Using the Real-Time Social Web." Springer, 2013.
Series	Lecture Notes in Computer Science
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/9953
Publisher's statement	The final publication is available at www.springerlink.com .
Publisher's version (DOI)	10.1007/978-3-642-37401-2_72

Downloaded 2024-06-23 05:56:02

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

Towards a Novel and Timely Search and Discovery System Using the Real-Time Social Web*

Owen Phelan, Kevin McCarthy, and Barry Smyth

CLARITY: Centre for Sensor Web Technologies
School of Computer Science & Informatics, University College Dublin
`firstname.lastname@ucd.ie`

Abstract. The world of web search is changing. Mainstream search engines like Google and Bing are adding social signals to conventional query-based services while social networks like Twitter and Facebook are adding query-based search to sharing-based services. Our search and discovery system, Yokie, harnesses the wisdom of the crowd of communities of Twitter users to create indexes of proto-content (or recently shared content) that is typically not yet indexed by mainstream search engines. The system includes an architecture [13] for a range of contextual queries and ranking strategies beyond standard relevance. In this paper, we focus on evaluating Yokie's ability to retrieve timely, relevant and exclusive results with which users interacted and found useful, compared to other standard web services.

1 Introduction

Yokie is a novel search and discovery service that harnesses the social content streams of curated sets of users on Twitter. The system aims to provide timely, novel, useful and relevant results by streaming content from Twitter. It also allows for partitioning of the Twitter social graph into topical or arbitrary groups of users (called *Search Parties*). While query-based search is at the core of both Yokie and traditional search services, our system is not designed or conceived to compete with or replace the latter. Yokie gathers its content from entirely different sources for a start, and this content is indexed in response to real-time search needs. As a result, the Yokie use-case of returning high-quality, real-time, breaking or trending content is very different from archival search approaches of traditional search engines.

Our work is motivated by the fact that the nature of online content is changing. The conventional web of links and authored content is giving ground to a web of user-generated content, opinions and ratings. At the same time, users who have previously relied on traditional search as their basic discovery service are finding more content through sharing on social networks. The advent of the social web has seen an important shift in the role of users from passive consumers of content to active contributors of content creation and dissemination.

* Supported by Science Foundation Ireland Grant No. 07/CE/11147 CLARITY CSET.

New technologies and services have emerged that allow users to directly manipulate and contribute diverse web content [7]. The Twitter microblogging service has, like the web as a whole, grown significantly in recent years [6][12].

While emphasis is placed on simple 140-character messaging, tweets can contain a wealth of contextual metadata relating to the time, location content and the user profile. Both Twitter and Facebook have become vital link-sharing networks [13][4]—our previous work has shown that approximately 22% of Twitter tweets contain a hyperlink string [13]. This prevalence of link sharing on social networks motivates an alternative approach to sourcing fresh, relevant and human-selected search engine content, instead of crawling the entire web-sphere. Also, by focussing on sourcing content from humans rather than the entire web, we can subdivide the landscape of content sharers into useful groups. This has numerous benefits, including the opportunity for users to curate their own ad-hoc lists, algorithmically group topical producers, and to aid in eliminating spam contamination.

A proof-of-concept prototype of Yokie was described and presented at [13], and a full system demonstration was presented at [14]. We have since added several user-feedback features to gain insight into the system’s usefulness, and propose a per-item global interestingness score; both of which are described below. In this paper, we are particularly interested in evaluating the performance of the system at providing timely, relevant and useful content. This is done by a live user trial, and an attempt at capturing comparative data with other established online systems.

1.1 System Background and Related Work

Information Retrieval and Search systems are rich and well-established research landscapes. The technique of Web Crawling [1] has been widely lauded and adopted, however it is computationally expensive and time consuming because it attempts to operate over the entire web. As such, some of the major research themes and challenges include introducing social signals for better relevance, and obtaining better content freshness in their indexes [4]. More recently, efforts have been made in applying social signals to search and content recommendation results [10][4]. For example, Dong et al. successfully exploit features of *recent* Twitter content to improve the performance of recency-sensitive queries of a standard portal search engine such as Yahoo [4]. Alternatively, recent work in retrieval has proven temporally-sensitive microblogs can be ranked and retrieved with reasonable precision and recall, while ignoring profile data altogether [9].

Yokie’s content is entirely mined from social streams. Increasingly, web users’ online time is spent mostly on social network activity [7]. Many of these services allow relatively open access to user content streams. It is no surprise then that many instances of recent literature includes analyses of Twitter’s real-time data and social graph, largely with a view to developing an understanding of why and how people are using such services; see for example [4][5][7][12]. The free and abundant access to the real-time web has also given rise to applications in other related fields beyond search. Phelan et al. describe useful recommendation

techniques for harnessing recently shared content from Twitter for a news discovery and recommendation service [12]. Chen et al. and Bernstein et al. have explored using the content of messages to recommend topical content [3]. Garcia et. al. successfully apply microblog data to product recommendation [5].

A main focus of applying social network properties to search systems should be on the users and contributors. Indeed, a core component of Yokie is the selection of *groups* of users for sourcing content (explained further below). Krishnamurthy et al. demonstrate that the user base of Twitter can be split into definitive classes based on behaviours and geographical dispersion [6]. They highlight the process of producing and consuming content based on retweet actions, where users source and disseminate information through the network. Several services have emerged that take advantage of the open access to profile data with a view to providing a score of a user's online activities (provided the user has given them permission). One such service, Klout (<http://www.klout.com>), aims to provide an engagement and influence score based on an aggregation of a user's activities across a broad set of online social networks. Other work has focussed on applying user-level reputation systems to social search contexts [10] and recommender systems [15].

Grouping users can have numerous benefits including the opportunity for users to curate or generate their own preferred lists of personal or topical sources from which to receive content. This would, in turn, allow for vertical searching across multiple domains [2]. This may be favourable compared to a catch-all attempt to gather all shared content. Previous work by [6][18] postulate a large and active subgraph of social networks like Twitter exist that exclusively publish spam. As such, any steps towards algorithmic subdivision of the social graph may need to consider the potential of irrelevant or spam-laden accounts.

1.2 The Yokie System

To better understand the system and our goals, we must briefly review the system architecture. Yokie brings together a number of key ideas to provide an alternative discovery service that emphasizes real-time, trending information rather than more static and archival content of traditional web search engines. The end goal is a service that allows users to query to find content from specified groups of users, with a variety of contextual querying and ranking functions.

“Search Parties” and Content Acquisition. A key aspect of the system is the curated list of content sources, what we've termed a *Search Party*. Yokie sources its information from topical communities of Twitter users. These communities are called *Search Parties*, and are intended to be either manually curated or automatically created. Links each member share are indexed in Yokie using the text of the Tweet with which it is shared. The system uses Twitter's API to acquire content as defined by the search parties. These messages are then stored and indexed using the service described in the following subsection. This component also carries out real-time language classification and finds other messages that contain the same URL so the system can calculate item popularity, and extracts tags so as to allow collaborative tagging.

Keyword + Time-window Query Fields

Query String	Date From	Date To	
instagram	1 week ago	now	🔍

Peeked Item Metadata
Shows original tweet + sharer, other users who have mentioned URL

Re-rank Menu

rank ↓

- relevance
- reputation
- interestingness
- newer first
- older first
- mentions
- longevity

0-30 / 119 results
in 4ms

"Peeked" Item

★ = Liked Item

Instagram for Android Got More Than 1M Downloads in 24 Hours
by LanceUlanoff + 375 others since 5 days ago
Instagram for Android has gotten more than 1 million downloads in less than 24 hours.
<http://mashable.com/2012/04/04/instagram-for-andr...> ★

Instagram • Instagram + Facebook
by themaria + 221 others since 2 hours ago
Instagram + Facebook - When Mike and I started Instagram nearly two years ago, we set out to change and improve the way the world communica...
<http://blog.instagram.com/post/20785013897/instag...> ★

Instagram For Android Hits 1m Downloads in Under 24 Hours
by adnys + 221 others since 5 days ago
<http://thenextweb.com/mobile/2012/04/04/instagram...> ☆

Twitpic - Share photos on Twitter
by dhH + 219 others since 1 hour ago
<http://twitpic.com/97ub0z> ☆

Facebook Buys Instagram for \$1 Billion - NYTimes.com

WHOA!!!RT @sonnygill: Wow - Facebook buys up Instagram. <http://t.co/mxPvbWz8>

Maria Ogneva
@themaria
Mon Apr 09 18:10:29 IST 2012

221 mentions

RT @instagram: We're excited to announce something very big: Instagram + Facebook <http://t.co/PBixZuF> by AnetaStepien

Damin Instagram is fugged RT @instagram: We're excited to announce something very big: Instagram + Facebook <http://t.co/MYrB84> by quencyty66

From the CEO of Instagram... <http://t.co/zUa27500> by tamermikhail

RT @instagram: We're excited

Fig. 1. Yokie’s UI includes functions for viewing extra explanatory metadata related to the item, and unary relevance ratings. This example resultlist is based on “links labelled Instagram in the past week by high-value technology people, ranked by popularity.”

Storage and Indexing. This subsystem is responsible for extracting meta-data regarding the tweets, for instance timestamp data, hashtags (#election2012, etc.), user profile information, location, etc. as well as the message content itself. The main content, the urlID of the URL mentioned in the message, and the timestamp is pushed to an indexer for storage and querying. In our current implementation we use Apache Solr (<http://lucene.apache.org/solr/>) for this. We also store the remaining extracted metadata in a MongoDB (<http://mongodb.org>) NoSQL database for easy retrieval and fast MapReduce functionality.

UI and Querying. The system is presented with a query interface, currently comprising of a query string field, and two temporal fields, *date from* and *date to*. The system takes as input a query string with an associated time window, which can be either a natural language query (eg. “1 day ago”, “now”, “last week”, etc) or a fixed date (“3 March 2013”), and then parses these into UNIX timestamps. The querying subsystem represents user queries, Q , based on the notation $Q=(\text{QueryString}, \text{SearchParty}, T_{max}, T_{min})$. The user can *click through* to visit a webpage and *peek* at an item, and drill-down on results to explore related content such as the original tweet that the URL was shared with, the time and date it was shared, and the related Tweet mentions (if any). Each item UI element contains a unary rating *like*, which users select when they believe an item is relevant to the query.

User and Item-Based Result Ranking. Users can rank using typical **Relevance**, which is vanilla Term Frequency Inverse Document Frequency scoring (TFxIDF) [16]. However, a set of ranking strategies beyond standard relevance have been devised using the added contextual metadata of the microblogs. These include ranking using temporal features such as item age (**Newer-First** and **Older-First**) and **Longevity** (the size of the time window between the first and last mentions of the given hyperlink). Popular items can be ranked as we capture a number of sample tweets that mention the hyperlink (labelled **Mentions** in the UI). **Potential Audience** is a user-based sum of follower-counts for each search-party member who have mentioned the link. The premise of this strategy is high-volume in-links for users can be a sign of high user reputation. It is easily possible to derive a range of user-based scores as we are specifically “listening” to people for content, as opposed to relying on a document graph that ignores content publishers.

We have recently implemented two new ranking strategies, both of which are part of our evaluation. **Klout** is a service that provides users of social networks an *influence score* based on user reach, engagement and their ability to drive other interactions. Using the Klout API, we can gather scores for each user (once Klout has a score computed for them). It is possible to rank content based on the *original publishers/sharers Klout score*. **Interestingness**, is an attempt to measure how interesting an item is based on activity of the item on the social network, and in other Yokie query sessions. Since we are privy to a multitude of content, context and user interaction data for each item, and much of this data is dependent on the conditions of the query, we can create a novel interestingness algorithm. Given the Query Q tuple, Interestingness is defined as:

$$Int(U_i, Q) = \left(\frac{Pop(U_i, Q)}{Lng(U_i, Q)} \right) \cdot \left(\frac{|Clk_{\forall U_i}|}{|Pk_{\forall U_i}|} \right) \cdot |Lk_{\forall U_i}| \quad (1)$$

$Pop(U_i, Q)$ and $Lng(U_i, Q)$ are the popularity and longevity of the current item U_i , given the parameters of the query tuple, (meaning their value is dependent on the query T_{max} and T_{min} values). $|Clk_{\forall U_i}|$, $|Pk_{\forall U_i}|$ and $|Lk_{\forall U_i}|$ represent the total number of clicks, peeks and likes for the item, irrespective of the parameters of Q . These values have a default value of 1 so as to avoid null values for interestingness of items with no user engagement. This algorithm utilizes the contextual features of the item, benefits from prior user engagement with the item, and ignores the profile data of the user. This allows the item’s score to be potentially independent of the original sharers’ explicit profile data.

2 Evaluating Yokie

In the following evaluation of Yokie we consider two main objectives. First, we explore how users engage and interact with Yokie, paying particular attention to their engagement with the different ranking strategies and their subsequent click-through rates. Second, we consider the potential for Yokie to succeed where

mainstream search engines fall short — as an exclusive source of timely or breaking content — by comparing Yokie usage to Bing Search and Bing News. In order to explore these evaluation objectives, we have conducted a live user trial of Yokie, to explore levels of engagement and interactivity. We use the data generated during this trial (queries and interaction data) as the basis for a comparative evaluation between Yokie, Bing Search and Bing News, focussing on the ability of the former to identify novel and timely content. Broadly, we have three main objectives we wish to evaluate from these two trials: **Objective 1** - Users *engage and interact with* user-based and contextually-ranked results. **Objective 2(A)** - Yokie locates *exclusive and timely results* compared with a mainstream Search Engine. **Objective 2(B)** - Yokie Users *engage and interact with* timely and exclusive content, compared with a mainstream Search Engine.

2.1 Open User Trial

We curated a Search Party of 1000 Twitter users whose profiles were designated as being *Technology* oriented. These were gathered by mining for names listed in 10 "Technology" labelled lists of users on the Listorious (<http://www.listorious.com>) Twitter list website. To garner high-quality users, each unique user found in these lists was assigned a Klout score that we used to determine the top 1000 users.

The search index was initially seeded by retroactively mining links from these user profiles and during the course of the evaluation these profiles continued to be mined for current links. As mentioned previously, the user interface allows users to perform three kinds of item-based interactions — namely click through (visit the webpage), "peek" for more metadata, and "like" using the star-shaped button beside each item. Users were also encouraged to re-rank results using the re-ranking UI. While queries retrieved an average of 50 results, users were presented with a maximum of 30 results at any given time. There was a possibility of being presented with different results when re-ranked, however we did not consider any particular ranking strategy as a benchmark to compare with others. Rather, we preferred to gain insight into the performance of the complete system. Each time the user began a new search session, a random ranking strategy was chosen so as to eliminate bias towards any individual options.

Summary Statistics. During the course of the two month trial, a total of 365,735 unique hyperlinks were mined from the Twitter profiles of our 1000 technology influencers. The user study attracted 125 unique users who submitted 1100 queries across 327 sessions. These queries led to the retrieval of 44,856 URLs, and in 95% of the cases, had a temporal window T_{min} set to "now" — indicating users' interest in up-to-the-minute results. Users interacted with 1061 items — 203 of which were click-throughs, and 858 were exploratory "peeks". While we encouraged users to click on the "Like" button for items, we found it difficult to elicit meaningful volumes of user feedback during the trial. As such, we will focus on user engagement of items based on Click-through and Peek actions. Further dataset and usage statistics are discussed below.

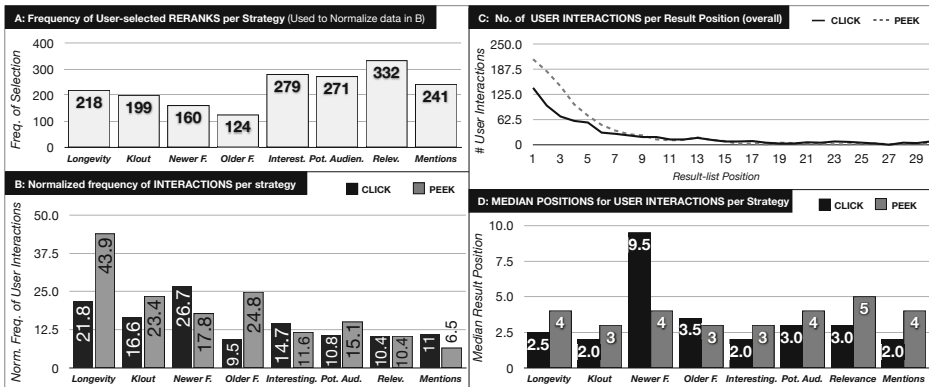


Fig. 2. Chart (A) shows frequency of user selected ranking strategies. Charts (B) shows normalized User Interaction Data for Clicks and Peeks. Charts (C) and (D) show Click Position Data for User Interactions.

Objective 1 — Users Engaged and Interacted with User-Based and Contextually-Ranked Results. Unlike more routine IR and Web Search evaluations which tend to focus on exclusivity of relevance-based rankings of results, in this work our aim is to explore the utility of different ranking strategies. As shown in Figure 2, users tended to actively test a variety of different strategies with a slight preference for strategies such as *Interestingness*, *Relevance* and *Mentions*. We use this data as a basis for normalizing per strategy user engagement of items.

Interactions vs. Ranking Strategies. While *Relevance* was the most frequently selected strategy by users, it performed poorly in terms of normalized user interactions. As presented in Figure 2 (B) users performed 2.6 times more click-throughs on items ranked by *Newer First*, and over twice as many items ranked by *Longevity*. Post-trial, we discovered that over 90% of the queries performed had a T_{max} of "now" (i.e. the query time). As such, it is conceivable the items selected in *Newer First* results lists were more topical or new to the user, or contained breaking news. Nevertheless, this is an interesting trend towards a preference for novel context-sensitive ranking. While "Peek" engagement of *Relevance* ranked items remains low, the trends are starkly different compared to the corresponding frequencies of selected ranking strategies. Users peeked at more items ranked by *Longevity* and *Klout* scores.

Result Positioning. Figure 2(C) represents median positions for Clicks and Peeks across all strategies. Both interaction methods follow a standard F-pattern of user interactivity as observed in other search result evaluations [11][8][17]. This chart presents a clear peak of user interaction for items on average above the 7th element in the Yokie result-lists. It also illustrates higher preference for users to perform exploratory "peek" actions on items instead of visiting the hyperlinks themselves. Finally, Figure 2(D) presents the median positions per strategy for

each interaction method. Here, we can interpret the positioning of items lower in the list as the system performing well at capturing interactions in the first few result items. In this case, Potential Audience, Klout, Interestingness and Mentions gained user click-throughs with a median position of the second item in the list, while the rest slightly higher. While it can be assumed that a better-performing engine would present higher-frequency interactions at the earliest position in the list, we can also argue the benefits of users exploring further beyond the top-n results. Recency-based (Newer-first) results garnered a higher median positioning — users explored more *recent items* lower in the list.

Discussion. There are a number of take-home messages that arise from these results. First, users do appear to be motivated to try a variety of different ranking approaches. There is a tendency to prefer those such as *Relevance* and *Interestingness*, at least with relation to their frequency of selection, perhaps because they align well with conventional Search Engine ranking. However, based on result engagement, we can see our novel strategies such as *Interestingness*, *Klout* and temporal strategies seem to outperform others.

2.2 Comparative Study with Bing

For every Yokie query performed during the open user trial, we also gathered search results from Bing Web and Bing News using a vector of the same *keyword queries* users submitted to Yokie (Bing’s API’s don’t support contextual features of the Yokie query tuple). For every Yokie query, we continued to collect Bing’s responses every 30 minutes so that we could track how these responses evolved over the evaluation period. In each case, Bing Web and News returned a maximum of 50 results, and each result item was stored based on its query, page title, timestamp of indexing by Bing, item positioning and description. Useful pieces of metadata returned by the API included a corresponding timestamp that represented the moment Bing captured or indexed the item, and the result-list position of an item with respect to that query in Yokie. These two features, along with the ability to compare and contrast overlapping content with Yokie allow us to do comparative studies to see if Yokie has gathered exclusive content, and whether it can present timely results higher up its result lists. Our assumption that the Twitter-sourced content would be newer and easier to index in a more timely fashion seems obvious, however little has been done to test it before.

Objective 2 (A) — Yokie Locates Exclusive and Timely Results Compared with a Mainstream Search Engine. Figure 3 illustrates dataset size and item overlap between the content captured from Bing Web, Yokie and Bing News. In the first panel, Figure 3(A), the comparison is based on the entire datasets of unique items captured by Bing Web and News using Yokie’s user trial queries, compared to the URL index of Yokie. In the second panel, Figure 3(B), represents the total coverage of user-interacted items compared to the Bing datasets. In each case, we pay particular attention to whether overlapping items appeared earlier in Yokie or Bing Web/Bing News, so we present a breakdown

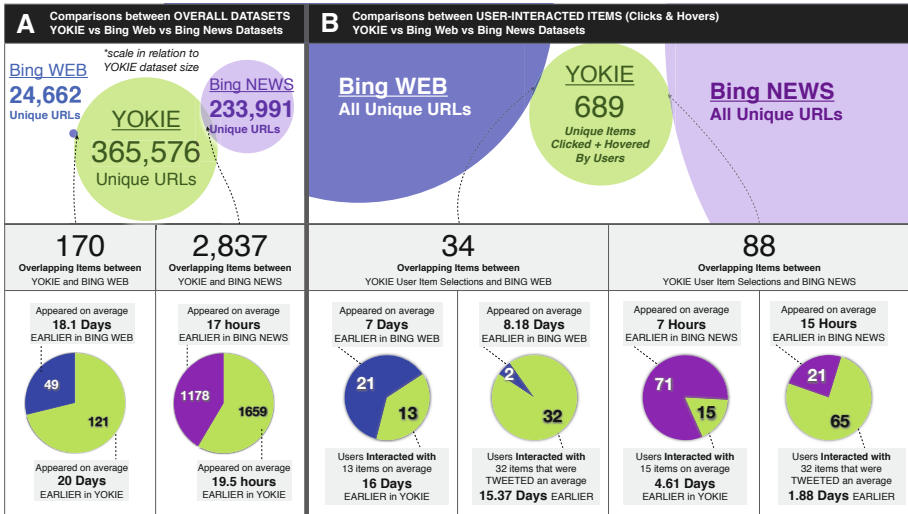


Fig. 3. Yokie dataset comparison with Bing Web Search and Bing News Search. Figure A shows a breakdown of item overlaps in the Bing Web / Yokie / Bing News datasets. Figure B shows the overlaps of user-engaged Yokie items and the total Bing datasets, and also highlight items that appeared earlier in respective datasets.

of the items which appeared earliest in either Yokie or Bing, and the average time of how *early* it was in the corresponding dataset.

Overall Dataset Coverage and Timeliness. As illustrated in Figure 3(A), we found only 170 Yokie results in the entire Bing Web result list — representing less than 0.1% of the total Bing Web items for the same queries. Of those 170 items, 121 (70%) appeared on average 20 days before they appeared in Bing Web. The remaining 49 items appeared earlier in Bing Web by an average of 18.1 days. When comparing the Yokie set with the arguably more timely of the comparative systems, Bing News, we see a much higher overlap set with 2837 items — almost 1% of the total News dataset. In this case, Yokie garners a considerably higher rate of coverage — 28% more of a share of those overlapping items earlier than Bing News. We believe that while gaining earlier items can be important, an overall broader coverage of items trumps Bing News’ gaining items on average 2 hours earlier across a smaller share.

Clearly Yokie is maintaining a very different type of information index compared to Bing (Web and News). It is less comprehensive than Bing Web, but it focusses on trending URLs and produces result-sets that are entirely different from those produced by Bing. The results returned by Yokie are exclusive to Yokie. If they have been indexed by Bing they are rarely retrieved, or swamped by more conventional results. If the results do come to be recognised by Bing, this usually happens after a significant latency period which can be measured in days for Bing Search and tens of hours for Bing News.

Item Exclusivity and Position Performance. Our next comparison is aimed at exploring how well Yokie performs at ranking and retrieving exclusive items that didn't appear in Bing. During the user trial, Yokie retrieved 44,856 of its 365,735 items from the 1,100 user-performed queries (not illustrated above).

If we compare this set of items with both Bing sets, we see negligible overlaps — 109 items appeared in any Bing Web result list, whereas 1,292 of them appeared in Bing News result lists. Therefore, Yokie's results contained 99.8% exclusive items compared to Bing Web, and 97.2% compared to the Bing News set (also of note is Bing News' 10 times larger set compared to Bing Web).

A more useful metric is comparing the Bing itemsets with items with which users interacted. To return your attention to Figure 3(B), we analyze a similar breakdown of overlaps as before, except here we compare them with overlapping items that are dated from when the user *interacted* with them. Also depicted is another interesting trend — of the 689 unique items, only a total of 122 overlapping items. Combined, this says that of all the items with which users interacted, 82% were *exclusively* present in Yokie's index.

Objective 2 (B) — Yokie Users Interacted with Timely and Exclusive Content. The previous objective was limited to analysing the overall composition of the datasets. Here, we examine the items explicitly selected by users from Yokie result lists, as these represent actual indications of user interest in the content. We separate each overlapping item based on whether its timestamp was earlier in Yokie or Bing. The timestamp can either represent the moment the user interacted with it, or the moment when the original item was published on Twitter (since Yokie live-streams content, conceivably its indexing time also).

Time of User Interaction vs. Bing Indexing. In Figure 3(B), we see that during the trial, 689 results were interacted with by Yokie users. Once again, the vast majority were exclusive to Yokie; only 34 items found by Bing web, and 89 by Bing News. In terms of timeliness, we also see Yokie performing comparatively well as opposed to Bing Web — 13 items were interacted with by users an average of 16 days earlier than they even appeared in the Bing Web index. Compare this to Bing Web's larger share of 21 items which appear on average only a week earlier in the Bing Web index than in Yokie's. Yokie is also seen to fall behind in terms of user engagement and Bing News publication. While Yokie users had clicked on items that were on average almost 5 days newer than their appearance in Bing News, this was only represented by 15 items. Bing News had a considerably broader coverage with 71 items with which Yokie users interacted, however these were in its indexes only 7 hours before Yokie. Overall, comparing the time of item interactions with Bing index time shows Yokie under-performing in terms of coverage of items with which its users interacted, regardless of its good performance gaining content earlier than Bing.

Interacted Items' Tweet Time vs. Bing Indexing. Here, we describe a similar comparison outlined earlier — but here we compare like-for-like analyses between the moment users *interacted with* an item, and its index time in Bing

Web and News. As described in Figure 3(B), Yokie users interacted with a total of 32 items that appeared on average 15.3 days earlier than they appeared in Bing Web. They also interacted with the majority of the news overlaps on average 1.88 days earlier than they even appeared in Bing News.

Discussion. One could sensibly assume that Bing Web and Bing News would garner higher coverage and timely content respectively compared to our system. However, the vast majority of Yokie results were *exclusive* to the system, apart from the 170 web items and 2837 news items that are overlapping. We can use this overlapping data to further support the hypothesis that Yokie is able to identify these results earlier than Bing — with over 70% of Bing Web/Yokie overlaps appearing almost 20 days earlier in Yokie, and almost 60% of the Bing News/Yokie coverage appearing nearly a full day beforehand in Yokie.

Once again, the data supports the argument for the potential of a real-time streaming technique for indexing web content. While Bing had more coverage of earlier items amongst user-interacted results, Yokie's interacted content was not only vastly exclusive, but also in many cases more timely than Bing. This is a surprising result considering Bing Web represents a capture of the entire web, and Bing News represents a real-time capture of topical breaking news.

3 Trial Discussion, Limitations and Conclusions

Our user trial and comparative evaluations have highlighted Yokie's potential for timely, exclusive and relevant content compared with popular web and news search services. Our approaches to contextual and user-based ranking strategies also performed well at providing relevant content with which users interacted. In particular, the temporally sensitive rankings that showed newer content and also content with high longevity garnered significant interest from users. Our second trial illustrates the shortcomings of both archival Search Engines and, surprisingly, some reputable online news search services at providing timely content.

This work is not without its limitations. In building Yokie we have designed a complex and novel information discovery service with many moving parts. This presented something of an evaluation challenge. Unlike traditional IR systems, there were no benchmark datasets (e.g. TREC [9], etc). Our approach has been to attempt a live user trial. Obviously in an ideal world one would welcome the opportunity to engage a large cohort of users — perhaps thousands or tens of thousands — in such an evaluation. Unfortunately this was not possible with the resources available, and so our trial is limited to a smaller cohort of just 125 users. Nevertheless these users did help to clarify our primary evaluation objectives and their usage provided a consistent picture of their common engagement patterns to show how they interacted with different ranking strategies and gained benefit from the results they provided. Moreover we bolstered this live evaluation with an additional study to compare Yokie with more conventional search and news discovery services by harnessing the activity of our live users as the basis for search queries. Our focus on Bing was purely pragmatic as it's API's Terms and

Conditions provided us with access to the data we needed, whereas alternatives such as Google do not.

Yokie's architecture and content source enables us to perform a broad range of future evaluations. Indeed, a multitude of ranking and retrieval techniques remain to be explored. Emphasis will be placed on trials in which users may curate their own Search Parties, and scope exists for these Search Parties to be collaboratively built. While the findings in this paper validate the system's merits and novelties, significantly more work can be done with the general retrieval aspects, such as the reasoning behind the comparatively lower positioning of user interesting items compared to Bing, comparative studies between the user-selected rankings and randomly assigned ranking strategies, and so on.

References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* 30, 107–117 (1998)
2. Chau, M.: Comparison of three vertical search spiders. *IEEE Computer* 36 (2003)
3. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Short and tweet: experiments on recommending content from information streams. In: *CHI 2010*. ACM (2010)
4. Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., Zha, H.: Time is of the essence: improving recency ranking using twitter data. In: *WWW 2010*, ACM, New York (2010)
5. Esparza, S.G., O'Mahony, M.P., Smyth, B.: On the real-time web as a source of recommendation knowledge. In: *RecSys 2010*, September 26-30. ACM (2010)
6. Krishnamurthy, B., Gill, P., Arlitt, M.: A few chirps about twitter. In: *WOSP 2008*, NY, USA. ACM (2008)
7. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: *WWW 2010*, pp. 591–600 (2010)
8. Lewandowski, D.: A framework for evaluating the retrieval effectiveness of search engines. In: *Next Generation Search Engines* (2012)
9. Macdonald, C., Ounis, I., Lin, J., Soboroff, I.: Microblog track. In: *TREC* (2011)
10. McNally, K., O'Mahony, M.P., Smyth, B., Coyle, M., Briggs, P.: Towards a reputation-based model of social web search. In: *IUI 2010*. ACM (2010)
11. Nielsen, J., Pernice, K.: Eye tracking research into web usability, <http://www.useit.com/eyetracking/> (accessed May 2012)
12. Phelan, O., McCarthy, K., Bennett, M., Smyth, B.: Terms of a feather: Content-based news recommendation and discovery using twitter. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) *ECIR 2011*. LNCS, vol. 6611, pp. 448–459. Springer, Heidelberg (2011)
13. Phelan, O., McCarthy, K., Smyth, B.: Yokie - a curated, real-time search and discovery system using twitter. In: *2nd RSWEB Workshop, RecSys 2011* (2011)
14. Phelan, O., McCarthy, K., Smyth, B.: Yokie: explorations in curated real-time search & discovery using twitter. In: *ACM RecSys 2012*. ACM (2012)
15. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. *Commun. ACM* 43, 45–48 (2000)
16. Sebastiani, F.: Machine learning in automated text categorization. *ACM Comput. Surv.* 34, 1–47 (2002)
17. Smyth, B., Balfe, E., Boydell, O., Bradley, K., Briggs, P., Coyle, M., Freyne, J.: A live-user evaluation of collaborative web search. In: *IJCAI* (2005)
18. Yardi, S., Romero, D., Schoenebeck, G., Boyd, D.: Detecting spam in a Twitter network. *First Monday* 15(1) (January 2010)