Automatic Identification and Contextual Reformulation of Implicit System-Related Queries

Adam Fourney
Microsoft Research
Redmond, WA, USA
adamfo@microsoft.com

Susan T. Dumais
Microsoft Research
Redmond, WA, USA
sdumais@microsoft.com

ABSTRACT

Web search functionality is increasingly integrated into operating systems, software applications, and other interactive environments that extend beyond the traditional web browser. In particular, intelligent virtual assistants (e.g., Microsoft Cortana or Apple Siri) often "fall-back" to generic web search in cases where utterances fall outside the set of scenarios known to the agent. In this paper we analyze a 3 month log of web search queries posed via the Cortana virtual assistant. We report that, in this environment, users frequently ask questions that implicitly pertain to the systems or devices from which they are searching (e.g., asking: [`how do I take a screenshot`]). Unfortunately, accurately answering these *implicit system queries* poses significant challenges to general web search engines, due in part to the lack of available context. We show that such queries: (1) can be detected with high precision, (2) are common, and (3) can be automatically reformulated to substantially improve retrieval performance in these fall-through scenarios.

CCS Concepts

•**Information systems** → **Query intent;** *Reformulation;*

Keywords

Implicit system search; virtual assistants

1. INTRODUCTION

Web search queries are often short and underspecified. To compensate, contemporary web search engines consider the contexts in which queries arise in an effort to better infer searchers' information needs [2]. For example, one can often expect to be directed to local establishments when searching [`where can I get sushi`], despite the query's failure to explicitly communicate a desire for geographic localization. Such queries are said to be implicitly localizable [11], and gains in retrieval performance can be achieved from strategies as simple as learning when to automatically reformulate requests to include mentions of a user's known location [4] (e.g., rewriting the former example to read: [`where can I get sushi in Pisa Italy`]).

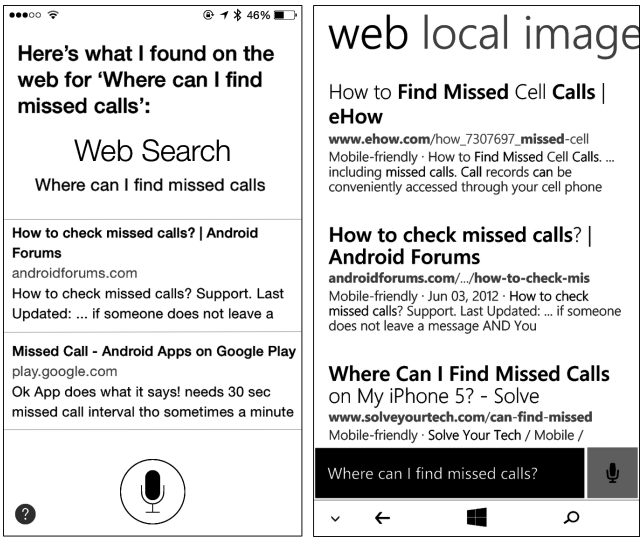

Figure 1: When asked [`where can I find missed calls`], both the iPhone virtual assistant (Siri, left), and Windows Phone virtual assistant (Cortana, right) elect to perform a web search. Lacking an awareness of system context, the search engine returns results for Android.

Increasingly, web search engines field queries from a range of new environments, including: operating systems [8], desktop applications [7], and virtual assistants (such as Siri [1], or Cortana [9]). Such integrations alter or extend the contexts in which queries may originate, thus posing new challenges in this space. For example, a user may ask [`where can I get sushi`] in one session, and [`where can I find missed calls`] in the next. This latter example, together with such queries as [`how do I mute this phone`], and [`set up voice mail`], typify a class of queries which implicitly pertain to the systems, platforms or devices from which they originate. In contrast to their implicitly localizable counterparts, these *implicit system queries* continue to suffer from their underspecificity, and web search engines often respond with results that are not relevant to their device context. For instance, iOS or Windows Phone users may be misdirected to resources about Android smart phones (Figure 1).

In this paper, we take inspiration from work done with implicit local search, and present methods for detecting and refining implicit system queries. The discussion is grounded

SIGIR '16, July 17-21, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2911451.2914701

by considering 3 months of spoken web search queries issued to the Bing.com search engine via the Cortana virtual assistant on Windows Phone devices. We begin by situating our work with respect to past research, then describe our dataset. We then separately consider two classes of implicit system searches, namely: *semi-implicit* system queries, in which a user's device is specified via a pronominal reference (e.g., "`can this phone be a hot spot`"); and *fully implicit* system queries, which lack such lexical cues (e.g., "`how do I enable a hot spot`"). The paper concludes with a discussion of future work.

2. BACKGROUND

To ground the remainder of the discussion, we consider web search queries posed to virtual assistants on smart phone devices. Virtual assistants allow users to engage in a conversational style of spoken interaction to access a device's commands, settings, and integrated services. Virtual assistants can also be asked to query the web, and often default to this behavior when a user's intent is unclear. Whether issuing commands or performing a web search, users are encouraged, by convention and instruction [1, 9], to structure their utterances as if they were talking with a person.

As we will show, this confounding of command & control functionality with web search, results in an environment where implicit system queries are common, and where web search results leave much to be desired. In the literature, there are two general strategies for coping with these challenges: Working from the top-down, recent research [5, 10] has sought to develop general classifiers that can detect *orphaned utterances* – queries in domains that one might suspect to be covered by virtual assistants, but which instead fall-back to a generic web search. Many of these orphaned utterances are device-related, and their detection can help to identify common ill-served intents.

Working from the bottom-up, earlier work has sought to leverage system context to improve the retrieval performance of underspecified system-related searches [3, 12]. For example, Ekstrand et al. demonstrate how software version information, together with details of a system's state, can be leveraged when evaluating web search queries related to image editing software [3]. These bottom-up efforts show promise, but assume all queries posed to the retrieval engine are system-related. This assumption is violated by intelligent virtual assistants that field queries about many topics.

The work presented in this paper represents a hybrid approach, striving first to detect implicit system queries, then to augment those queries with contextual details of the user's system. As noted in the introduction, our approach is most similar to work done in implicit local search [4, 11].

3. DATASET

Our dataset consists of 3 months of spoken queries issued to the Bing.com search engine via the Cortana virtual assistant. The dataset contains tens of millions of English language queries, originating from within the United States, collected from July 1st, 2015 through September 30th, 2015. Prior to analyses, queries were normalized by removing punctuation, and by converting text to lowercase. Additionally, compound nouns containing the word "`phone`" were normalized as follows: Compound nouns synonymous with phone, such as "`telephone`" or "`mobile phone`", were converted to the base word "`phone`". The remaining phone-related compound nouns were tokenized such that each appears as a single term (e.g., "`phone number`" becomes "`phone_number`", "`phone bill`" becomes "`phone_bill`", etc.) This normalization facilitates phrase-based matching strategies, which, as we show in the next section, are an effective means for detecting semi-implicit system queries.

4. SEMI-IMPLICIT SYSTEM QUERIES

To motivate our further study of implicit system queries, we consider the subset of queries that betray their system-related nature, but which otherwise leave implicit the identities of the systems to which they pertain. Examples, include: [`can this phone be a hot spot`], [`how to silence the phone`], and [`set my phone's quiet hours`]. We refer to such queries as *semi-implicit*. In the remainder of this section, we demonstrate that semi-implicit queries can be reliably detected with simple heuristics, that they occur frequently in the logs, and that simple query alteration strategies can substantially improve retrieval performance.

4.1 Detection and Alteration Heuristics

We identify semi-implicit queries heuristically by identifying searches that: (1) have been posed by more than one user, and (2) contain indicator phrases such as "`my phone`", "`this phone`" or "`the phone`". The first condition, which restricts analysis to non-unique queries, serves as a heavy-handed filter for excluding searches which may have been issued accidentally (e.g., pocket dialing), or which may contain speech recognition errors. Roughly half of the distinct queries in our dataset are excluded by this filter.

In addition to detecting semi-implicit queries, we explore two reformulation strategies that aim to improve the quality of retrieved results. The first strategy replaces indicator phrases with the model names of the users' smart phones, as determined from client meta data (e.g., the "`UserAgent`" header). Alternatively, the second reformulation strategy replaces indicator phrases with the names of the devices' operating platforms (e.g., "Windows Phone", "iOS", etc.) Finally, we consider a third approach which directly interleaves the results retrieved by the first two alterations. The merged results alternate between device- and platform-specific responses (with duplicates removed).

4.2 Precision of Indicator Phrases

To estimate the precision of our detection criteria, we randomly sampled 100 queries identified by each of the indicator phrases listed in Table 1a. We then presented crowd workers with a contextualized scenario, and tasked them with labeling the results. An example labeling task is as follows:

> *Sarah activates her smart phone's voice interface and says: "how do I silence the phone". You observe that Sarah is using a Lumia 640 smart phone, which runs the Windows Phone operating system. When Sarah spoke the phrase "the phone", was she talking about her Lumia 640?*

We obtained 5 judgments for each of the 300 queries, and used a majority voting criteria to derive a final set of labels. We find that, within our sample, the precisions of the heuristic classifiers ranges from 0.89 to 0.96 (Table 1a). Moreover, we inspected queries labeled as false positives,

(a) semi-implicit system queries	precision
containing "*my phone*" (sample N = 100)	0.96
containing "*this phone*" (sample N = 100)	0.90
containing "*the phone*" (sample N = 100)	0.89

(b) fully implicit system queries	precision
queries selected via LR test (sample N = 300)	0.90

Table 1: Precision of semi-implicit (a) and fully implicit (b) indicator phrases, as judged by a panel of independent human labelers.

and found that such searches were frequently difficult to interpret (e.g., [`look the phone`], and [`ok my phone`]). Such queries may constitute non-unique speech input errors or false activations.

4.3 Impact to Search Traffic

Applying the query detection heuristics to the complete 3 month query log reveals that users are 1.52 times more likely to refer to their devices with semi-implicit phrases like "`my phone`", or "`the phone`", than to *explicitly* mention the type of device they are using. (Table 2a,c). Moreover, when semi-implicit phrases were used, searchers were only about half as likely to click on a search result. These differences are highly statistically significant (χ^2 goodness of fit, $p \ll 0.0001$), and, together, strongly support this paper's focus on improving the retrieval performance of implicit system queries.

4.4 Retrieval Performance

Finally, we characterize the retrieval performance of the automatic query alteration strategies outlined above. For each of the 300 sampled queries (i.e., those mentioned in section 4.2), we retrieved web search results with: the unaltered query, the device-specific alteration, and the platform-specific alteration. Crowd workers again provided judgments – this time rating relevance on a 5-point scale ranging from Poor to Perfect. We focus on the top three search results in each condition, as the phones in our dataset could only display two or three results at a time (Figure 1). Table 3a reports the mean nDCG@3 for the various conditions, as well as the number of individual queries whose nDCG@3 scores were improved (wins) or hindered (losses) by the alteration. We found that, on average, reformulations outperformed unaltered queries across all treatment conditions. Moreover, we found that between 66% - 82% of individual queries benefited from reformulation (only 13% - 22% were impacted negatively). All win/loss ratios reported in table 3a are highly statistically significant (sign test, $p \ll 0.0001$). In more practical terms: across all queries, the median relevance of the *topmost* search result was 4 (excellent) in all treatment conditions, but was only 2 (fair) for the unaltered query baseline. These findings speak to the efficacy of simple, but targeted, reformulation strategies in this context.

(a) semi-implicit system queries	relative search volume	relative CTR
contains "*my phone*"	1.08	0.54
contains "*this phone*"	0.26	0.64
contains "*the phone*"	0.19	0.24
any of the three	1.52	0.52
(b) fully implicit system queries		
queries selected via LR test	8.68	0.45
(c) explicit system system queries		
mentions device's operating platform	0.73	1.04
mentions device's hardware model	0.28	0.89
either of the two	1.00	1.00

Table 2: Normalized search volume and click-through rates (CTR) of queries classified as (a) semi-, or (b) fully implicit system queries in the 3 month dataset. Values are normalized with respect to queries that *explicitly* mentioned the operating platform or hardware model of the device from which they originated (c).

5. FULLY IMPLICIT SYSTEM QUERIES

We now turn our attention to *fully implicit system queries*, which provide no overt indication of their system-related nature. Examples of fully implicit system queries include [`how do I take a screenshot`], and [`change the wallpaper`]. We present a fully unsupervised log-based approach for detection and alteration of such queries.

5.1 Detecting Fully Implicit System Queries

When faced with unsatisfactory search results for implicit system-related queries, users often reformulate their searches by adding additional details of their system context. Mir-

	Mean nDCG@3				# Wins / No Change / Losses		
(a) semi-implicit system queries	**no alteration**	**platform**	**device**	**interleaved**	**platform**	**device**	**interleaved**
containing "*my phone*" (N = 100)	0.43	0.55	**0.69**	0.64	66/14/20	**80/4/16**	77/3/20
containing "*this phone*" (N = 100)	0.39	0.54	**0.71**	0.66	67/13/20	**82/5/13**	76/3/21
containing "*the phone*" (N = 100)	0.39	0.52	**0.66**	0.63	66/12/22	78/3/19	**80/1/19**
(b) fully implicit system queries							
queries selected via LR test (N = 300)	0.57	0.68	0.68	**0.69**	**180/44/76**	181/32/87	196/17/87

Table 3: Retrieval performance for various reformulation strategies, applied to semi-implicit (a) and fully implicit (b) system queries. The leftmost columns report mean nDCG @ 3. The rightmost columns report the number of individual queries whose nDCG @ 3 score increased (wins), remained unchanged, or decreased (losses) after reformulation. All win/loss ratios, in both (a) and (b), are highly statistically significant (sign test, $p \ll 0.0001$).

roring past work [6], we leverage the likelihood ratio test to detect these non-accidental reformulations. We define a *query pair*, $\langle q_1, q_2 \rangle$, to be a pair of successive queries, issued by the same user, occurring no more than 30 minutes apart in the log dataset. The second query, q_2 is then abstracted as a boolean event, s_2, which occurs whenever q_2 mentions a user's system, platform or device by name. We then consider the goodness of fit of two competing hypotheses that explain the observed occurrence frequencies of $\langle q_1, s_2 \rangle$ pairs:

$$H_0 : P(s_2|q_1) = P(s_2|\neg q_1)$$
$$H_1 : P(s_2|q_1) \neq P(s_2|\neg q_1)$$

The first hypothesis, H_0, asserts that s_2 occurs independently from q_1. Conversely, H_1 asserts a dependence; and, in practice, we find that $P(s_2|q_1) >> P(s_2|\neg q_1)$. The likelihood ratio test statistic, $\chi^2 = -2\ log\ (L(H_0)/L(H_1))$, can be computed directly from observed data, and is known to be asymptotically Chi-squared distributed, with 1 degree of freedom [6]. In this paper, we reject H_0 when the test statistic is larger than 28.00. This corresponds to a p-value $< 10^{-7}$, and is strong evidence that q_1 requires an explicit system reformulation. Once discovered, such queries are placed in a lookup table used to identify future instances.

5.2 Empirical Results

As before, we report values for precision (Table 1b), traffic impact (Table 2b), and retrieval performance (Table 3b). By many measures, implicit system searches perform similarly to semi-implicit queries. Notably, classifier precision is again judged to be 0.90, and the query alteration strategies again lead to mean nDCG@3 values approaching 0.7 in all three conditions. For implicit system queries, reformulation improved between 60% - 65% of all individual queries (wins), and negatively impacted 25% - 29% of queries (losses). The slightly lower win/loss ratio, compared to semi-implicit queries, can be attributed to the higher baseline performance in the fully implicit case. The chief distinction between semi- and fully implicit system queries, however, is one of traffic volume: We find that fully implicit system searchers occur 5.7 times more frequently than semi-implicit searches, and 8.6 times more frequently than those that explicitly specify a device or platform.

6. CONCLUSION & FUTURE WORK

When users speak queries to intelligent virtual assistants, their utterances often implicitly pertain to the devices from which they originate. In this paper, we have we have shown that such queries: (1) can be reliably detected, (2) account for a meaningful proportion of search traffic, and (3) can be reformulated to greatly improve retrieval performance. Nevertheless, our analysis is limited in several ways, and there are numerous opportunities for future research. We elaborate below.

Citing a desire to filter accidental query activations and speech recognition errors, our methods consider only queries that have been issued by more than one person. However, we believe there is little risk in lifting this restriction, as user are unlikely to attend to search results in cases of false activation. Further study is necessary to more fully investigate this issue.

We are also sensitive to the possibility that users may use virtual assistants to query about *other* devices or platforms, and that the proposed reformulation strategies may hinder such efforts. In our log dataset we found that users *explicitly* query about their own devices roughly 4 times as often as they query about other platforms. We hypothesize that this difference is even more dramatic for implicit system queries – especially in scenarios where queries contain phrases such as "`this phone`", or "`my phone`". Again, further study could help to elucidate these factors.

We also recognize that, just as there are degrees of geographic locality (e.g., country, state, county, address), so too are there degrees of system locality (platform, product line, device model, etc.) On average, our results suggest that reformulations should be as specific as possible (i.e., prefer model over platform). However, there are individual cases where more general reformulations yield superior results. This suggests opportunities for the development of more nuanced classifiers that can determine the type of reformulation needed to best address an implicit system query.

Finally, while we focus on queries issued to virtual assistants, we feel that implicit system searches will become an important class of queries moving forward, especially as web search is integrated into new applications and environments.

7. REFERENCES

[1] Apple Corporation. iOS - Siri - Apple. http://www.apple.com/ios/siri/, 2015.

[2] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the Impact of Short- and Long-term Behavior on Search Personalization. In *Proc. SIGIR '12*, pages 185–194, New York, NY, USA, 2012. ACM.

[3] M. Ekstrand, W. Li, T. Grossman, J. Matejka, and G. Fitzmaurice. Searching for Software Learning Resources Using Application Context. In *Proc. UIST '11*, pages 195–204, New York, NY, USA, 2011. ACM.

[4] L. Gravano, V. Hatzivassiloglou, and R. Lichtenstein. Categorizing Web Queries According to Geographical Locality. In *Proc. CIKM '03*, pages 325–333, New York, NY, USA, 2003. ACM.

[5] D. Hakkani-Tur, Y.-C. Ju, G. Zweig, and G. Tur. Clustering Novel Intents in a Conversational Interaction System with Semantic Parsing. In *Proceedings of Interspeech*. ISCA, Sept. 2015.

[6] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating Query Substitutions. In *Proc. WWW '06*, pages 387–396, New York, NY, USA, 2006. ACM.

[7] Microsoft Corporation. Do things quickly with Tell Me. https://support.office.com/en-us/article/Do-things-quickly-with-Tell-Me, 2015.

[8] Microsoft Corporation. Search for anything, anywhere - Windows Help. http://windows.microsoft.com/en-us/windows-10/getstarted-search-for-anything-cortana, 2015.

[9] Microsoft Corporation. What can I say to Cortana? http://www.windowsphone.com/en-us/how-to/wp8/cortana/what-can-i-say-to-cortana, 2015.

[10] G. Tur, A. Deoras, and D. Hakkani-Tur. Detecting Out-Of-Domain Utterances Addressed to a Virtual Personal Assistant. In *Proceedings of Interspeech*. ISCA, Sept. 2014.

[11] M. J. Welch and J. Cho. Automatically Identifying Localizable Queries. In *Proc. SIGIR '08*, pages 507–514, New York, NY, USA, 2008. ACM.

[12] J.-R. Wen, N. Lao, and W.-Y. Ma. Probabilistic Model for Contextual Retrieval. In *Proc. SIGIR '04*, pages 57–63, New York, NY, USA, 2004. ACM.