What's the intention behind your query? A few observations from a large developer community

Alexander Löser, Wojciech M. Barczyński, Falk Brauer

SAP AG
SAP Research CEC Dresden
Chemnitzer Str. 48e
01187 Dresden, Germany
{alexander.loeser,wojciech.barczynski,falk.brauer}@sap.com

Abstract. We study common query intentions in a software developer network with more than one million users. Based on a large query log analysis we could identify typical search intentions and identify common entities. For resolving most frequent query intentions and to identify entities and relationships from relevant pages we recommend state-of-the-art information extraction technologies.

1 Introduction

Corporate portals often are the dominant source of information for employees, customers and other company affiliates. E.g., SAP's software developer network (SDN) at *http://sdn.sap.com* is SAP's premier community site for SAP developers, experts and software engineers. In such portals users have access to documents about products by browsing a predefined static taxonomy of software systems, or may use a full text search option, which is based on keywords. One current problem is a lack of precision for keyword based search queries. Table 1 shows the top 10 queries for March 2007 for SDN. Even though the correct result for all of the queries is available via the SDN portal, none of the right results was mentioned in the first three answer pages (or 30 result links) from the search engine. However, the ultimate goal of any search system is to answer the intention behind the query [1]. Recent research on Intranet search technology established that for transactional queries e.g., download requests for software [13], and navigational queries e.g., home page search [21], precise answers could be derived using information extraction techniques.

However, developers of search portals seldom have a clear understanding about common queries and potential answer pages from sources of consumer generated content. E.g., if a source mostly contains high quality information, how often the information changes, what is the social process the content is created etc. Based on a detailed query log analysis we study common query intention classes for navigational, informational and transactional queries for a corporate portal. Our contribution is to identify common intention classes and suggest extraction technologies for each class. We believe that our analysis is general approach for corporate portals e.g., *help.sap.com*, *sap.ittoolbox.com* or *msdn.microsoft.com*. To summarize:

- We give a detailed query log analysis including frequent and infrequent queries over a period of four weeks in May 2007.

– We analyze quantitative characteristics and unravel the most popular user intentions.
– For the four most common query intention classes we recommend state-of-the-art information extraction techniques to obtain correct answers.

This paper is organized as follows: in Section 2 we give a detailed analysis on query logs. In Section 3 we unravel common user intentions. Section 4 concludes with related work.

Table 1. Top 10 queries

Query	**Frequency**	[%]
ruby	40,098	2.36
firebug	12,546	0.74
solution manager	2,070	0.12
workflow	1,515	0.09
abap	1,455	0.09
wiki	1,282	0.08
visual composer	1,270	0.07
smartforms	1,217	0.07
idoc	939	0.06
alv	928	0.05
XI	816	0.05

2 Query log characteristics

We use a sample from SDN query logs of May 2007, including 470.973 total number of queries and 12.609 unique entries (Table 2). Unique queries have been identified after a normalization process including trimming white spaces at the beginning and the end, lowercasing query terms and excluding empty queries. After normalization, unique

Table 2. General query logs characteristic

Feature	Value
Total number of queries	470973
Number of empty queries	62603
Unique query before normalization	14408
Unique queries after normalization	12608

12.608 queries remained. In total, they have been submitted 408.370 times (Table 3). The distribution of queries is similar to logs from Internet search engines in the sense that they both have a long-tail distribution for query frequency [17]: A very few queries are very common, but most of the workload is on queries that individually occur very

rarely. To provide an accurate measure of user intention study, we created two subsets: Q_1 includes the top 200 most frequent queries (Table 4) and Q_2 includes 200 queries randomly chosen from the less frequent queries (Table 5). The probability for choosing an infrequent query was counted from its frequency divided by the sum over all other infrequent queries. Table 3, 4 and 5 introduce common query characteristics. We derive the following observations:

- **One keyword queries are highly common.** 68% of the most common and 40 % of the less common unique queries are one keyword queries.
- **Optimizing for top one keyword queries boosts precision.** Only by optimizing the search engine to the top 137 one keyword queries, the search system could give an exact answer to nearly one quarter of the search requests (cf. Table 4).

Table 3. All queries - number of terms

# terms	# total	[%] total	# unique	[%] unique
1 kw	215,573	53	3,806	30
2 kw	111,539	27	4,033	32
3 kw	45,099	12	2,455	20
4 kw	18,309	4	1,166	9
>4 kw	17,850	4	1,148	9
Avg. terms	1,857			

Table 4. Top 200 frequent queries - number of terms

# terms	# total	[%] total	# unique	[%] unique
1 kw	111,669	83	136	68
2 kw	20,845	16	55	27
3 kw	1,754	0.7	7	4
4 kw	0	–	0	0
>4 kw	419	0.3	2	1
total	134,687		200	

3 Most common user intentions

Recent research [1] has shown the importance of understanding the query intention. In this section we first define classes of query intentions. We analyse how often users a request matches a particular intention class and map intention classes to common extraction technology.

Table 5. 200 infrequent queries - number of terms

# terms	# total	[%] total	# unique	[%] unique
1 kw	3982	45	80	40
2 kw	3034	33	59	30
3 kw	1359	15	36	18
4 kw	419	5	18	9
>4 kw	145	2	7	4
total	8939		200	

3.1 Relevant query intention classes

Table 1 shows a sample of the top 10 queries. Except for *workflow*, all are of the type navigational queries.

Motivated by this observation we manually investigated the top 200 queries and less frequent 200 queries. Given a particular query, we determined the most relevant answer for the query in a gold standard. For spotting the answer documents we used the current SDN search engine as initial seed answers and conducted further browsing. For most queries only one answer was found. However for few queries we also spotted two or more answers. Following [16] we structured our query set into navigational, informational and transactional queries. However, for the special case of a software developer portal we redefined the following query classes:

Navigational queries. Navigational queries are directed towards navigating three different classes of web pages:

- **Product sites.** Typical queries are *[solution manager],[visual composer]* with the intention to navigate towards the product web page or visit this site again. Sometimes also abbreviated product names are used, such as *[XI]* for *[exchange infrastructure]*.
- **Sub sites.** These queries intent to visit sites related to the structure of the portal. Example queries are *[wiki], [blogs]*. The intention of these queries is to navigate to site directly using a search query instead of browsing the link structure.
- **Developer notes.** The portal publishes 'developer notes' describing solutions to specific problems. They are identified by a six to seven digit number. Such numbers are infrequently used as search request e.g., *[701654]*.

Informational queries. One of the portals major goals is to support software developers with advice and support for common and rare software problems. Another goal is to form a community and to present SAP's product portfolio. Both tasks, listing products and resolving problems, are typical candidates for informational queries. From our sample we could discover several types of informational queries:

- **Closed queries using question words.** Users submit these queries and intend to find web pages for solving a specific problem. Common question words, such as "how to" or "what is" are used. E.g., an example query is *[How to create an XSLT mapping file]*.

- **List queries.** These queries intend to list common products given a technical concept without explicitly mentioning (or even knowing) the name of a particular product. E.g., the query *[data mining]* intents to list data mining products like *SAP Business Intelligence suite* or *SAP Accelerator*.
- **Advice queries.** Typical advice queries request instructions for installing or configuring software products, such as *[solution manager configuration]*.
- **Locate queries.** Often customers "copy & paste" messages directly from an applications or from code files. Query examples are (error) messages, such as *[OBJECTS_OBJREF_ NOT_ ASSIGNED]*, code fragments, such as *[MESSAGE_ TYPE_X]* or request information on configuration parameters, such as *[login/create_sso2_ticket]*. The search goal of such queries is to locate technical documents or relevant forum threads, where the query request is mentioned.

Transactional queries. The portal supports typical transactions for software products, e.g. users may download or upgrade software. In this study we could only count those download requests where potential software was available for download. We assume that more queries intent towards downloading software.

Unclear queries. Unfortunately, not every query could be mapped to one intention class. We mapped a query to the unclear query class, if at least one of the following conditions was true:

- **No clear intention.** For queries like *[jdbc]*, *[performance]* or *[install]* we could not identify a clear intention.
- **Ambiguous intentions.** Some queries could be mapped to more than one intention classes. E.g., the query *[widget]* could refer to a *list query* - informational; or a *product site* (SAP widgets) - navigational.

3.2 Query intention distribution

Table 6 shows detailed information about common query intentions. In contrast to a previous study [21], where navigational queries are the most dominating class among the frequent queries, in this study navigational and informational queries are nearly equally often issued. Furthermore, informational queries are the most often requested class of infrequent queries. Transactional queries (at least for software products) appear quite infrequently. One explanation might be that the portal provides many documents about help and problem solving, but only few software downloads. We focus our analysis to the following three most requested query classes:

1. **Informational "What is" and "How to" queries.** Users frequently use closed queries to address a request. 215 queries refer to this most frequent query intention class.
2. **Navigational queries for products.** 85 belong to navigational queries for product home pages. We could confirm the findings of [21]; both, long and abbreviated forms, are equally often used.
3. **Informational queries listing technical concepts.** 60 queries are related to list further information for given technical concept. These queries are slightly more issued among the frequent queries than among the infrequent queries.

Table 6. Detailed common search intentions

Intention	Top 200	Less 200
Navigational		
Product home pages	41	9
Abbr Product home pages	34	1
Sub sites	2	3
Developers Notes	0	1
Informational		
What is/How to	36	73
List technical concepts	33	27
(Configuration) Advice	1	12
(Error) Messages	2	8
Code Fragments	4	3
Locate documents	2	10
Configuration parameters	0	3
Transactional		
Download request	9	2
Unclear		
	36	48

3.3 Which queries could we resolve?

Unfortunately, building portals based on intention recognition technology remains a time and cost intensive project. First, common intentions need to be unraveled. Next, possible answer sources need to be identified. Third, for each potential source of answers, specific information extraction technologies need to be developed to relevant catch entities and relationships from documents. Last, user intentions and content sources change over time, thus the search engine not only needs to be adjusted but also extend to new search intention and content sources. Today, there is no 'best practice' on how to best ramp up and maintain such search portals. In particular, we identified the following shortcomings of the current search solution:

1. **Poor entity and relationship recognition capabilities.** Search results are a presented as a list starting with the most relevant result. Search relevance is computed using traditional TF/IDF ranking techniques, without recognizing entities and relationships. However, the user expects that the search engine would recognize relevant entities and its relations in search queries documents. Specifically, relevant entities could be relationships between product entities and error message entities or between incompatible products.
2. **Monolithic search engine architecture.** The current search architecture is based on a monolithic approach (cf. Section 4). It includes different services, such as preprocessing, indexing and searching. For understanding entities and relations this "one size fits all" approach is costly to maintain and to extent.
3. **Insufficient distinction of source quality.** Most of the search portals integrate different sources (or sites) which differ in quality, moderation and presentation of the content. We distinguish between sub sites that are generated by the users e.g., *wikis*

or *forums*, and content provided by professional authors, such as the *product home pages*. The current search engine does not distinguish between these sources, their different content quality or content production process.

To address these challenges we give some preliminary solutions on how to resolve different query intentions. Given table 6 we focus on navigational and informational queries. By applying state-of-the art technology in this section we suggest approaches for resolving the intention for more than half of the queries in our sample of 400 queries.

Navigational queries. We focus on the following query intention classes representing 91 navigational out of 400 unique queries:

- **Queries for product home pages and sub sites.** We observed that web page authors mark a page as navigational by using a discriminating term, e.g. (abbreviated) products, error messages, or names for sub sites of the portal. Furthermore in *wikis*, *forum* and the main portal these entities appear in the URL and the title of the page. For resolving such entities named entity recognition (*NER*) technology (e.g. based on simple list based entity extraction methods as defined in [4]) or more complex rule based approaches as defined in the AVATAR project [7, 19] are common. Such navigational queries are resolved by spotting and indexing navigational pages *a priori* [21, 5]. For each spotted navigational page n-grams titles, URLs and anchors, are extracted and stored in a separate index. A query is matched against the index and the top results of the most relevant types are returned. If the query does not match this special-case-index, results from the organic search engine are returned.
- **Developer note queries.** These queries aim to spot a particular document given a unique identifier of the document. Our approach is to recognize first if a page is a developer note, e.g., by spotting the terms *developer note* in the title and check for following six digit number identifying the note. We extract the ID from the document and match queries against an index of all recognized developer notes IDs.

Informational queries We focus on the following query intention classes representing 119 informational out of 400 unique queries:

- **What is/How to queries.** Purpose of asking *What is* queries is getting an definition of a product or technology. From our gold standard (cf. section 3) we observed that 19 *What is* queries often could be answered with a *wiki* document. 14 *How to* queries are resolved with a *forum* page and 62 in the *Wiki*. E.g., for query *[lo extraction step by step]* the answer can be found at thread title *Lo extraction*. Because the *forum* and the *Wiki* are moderated we expect a higher quality and more consistent structure on these pages.
- **System messages queries.** Queries of this type include error and system messages. To create a query request, most users just copy a system message to the search engine to locate more information e.g., why this message has been thrown. Another interesting observation is that these queries have an interesting discriminating feature: In our sample the messages either had a length of more than five keywords,

such as *[A pop-up window was blocked in visual composer]* or a length of one keyword, such as *[DATASET _ WRITE _ ERROR]*. Please note that we expected the relevant answer in a document covering all the error codes. However we observed that for most of these queries the relevant answer was discussed in a thread of the forum.

4 Related Work

There are four broad research areas of work that are relevant to the work presented in this paper. The following section discusses related work.

Understanding search goals. The classification of search queries into navigational, transactional, and informational was originally proposed in [1]. Several examples and scenarios for each class of queries in the context of enterprise search are described in [6] and a more recent analysis of user goals in Web search is presented in [16]. There has also been prior work in the use of techniques based on classification and user behavior for automatic user goal identification [9, 10, 12].Transactional queries in the Intranet has been investigated in [13, 8]. Analogous to our approach of pre-identifying and separately indexing navigational pages, the work presented in [13] describes a similar process for the class of transactional queries.

Intranet search. Upstill et. al. [20] investigate the use of evidence such as in-degree, variants of Page-Rank, and URL-type, when identifying home pages on several test collections including an intranet data set. Their results indicate that of the three types of evidence investigated, re-ranking based on URL-type provided the maximum benefit. The study on "workplace web search" by [5] established that several conventional ranking approaches that find favor in Web search are effective discriminators when applied to intra net pages. The authors of [2] also elucidate the differences between search systems for the Web and those designed for enterprises. How to resolve navigational queries in the intranet was studied by the authors of [21]. Their approach is based on off-line identification of navigational pages, intelligent generation of term variants to associate with each page, and the construction of separate indices exclusively devoted to answer navigational queries.

Web Page Search. There is a large amount of work in the area of using structural information on a Web page (such as URL, anchor text, and title) to improve general Web search and link-based page classification [3, 8, 11].

Text Analytics. Text analytics is a mature area of research concerned with the problem of automatically analyzing text to extract structured information. Examples of common text analytic tasks include *entity identification* (e.g., identifying persons, locations, organizations, etc.) [4], *relationship detection* (e.g., person X works in company Y)[15, 18] and *co-reference resolution* (identifying different variants of the same entity either in the same document or different documents) [14]. Text analytic programs used for information extraction are called **annotators** and the objects extracted by them are called

annotations. Traditionally, such annotations have been directly absorbed into applications. A prominent example is the AVATAR Information Extraction System (IES) which tackles some of these challenges [19].

5 Conclusions

A major part of corporate search portals currently are not able to analyze the user intention and thus often confront the user with imprecise answers. Based on a large query log for a corporate portal we could identify common search intentions e.g., for product home pages, sub sites as common navigational query intentions and *What is* and *How to*, system messages queries as common informational queries. State-of-the-art information extraction technologies are able to identify entities and relationships from relevant pages.

Our work has only scratched the surface of potential research questions: Could we apply our study to other sources of unstructured information than the corporate web e.g., to customer relationship management systems or product information systems? Could we use existing corporate structured data to discover entities and potential relationships between them? How could we share extraction knowledge and the extracted information itself? It is our hope, that by applying the operators to different company scenarios we will improve the abstraction level and inspire a large community to write and share efficient implementing for new and existing operators.

References

1. A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.
2. A. Z. Broder and A. C. Ciccolo. Towards the next generation of enterprise search technology. *IBM SYSTEMS JOURNAL*, 43(3):451–454, 2004.
3. N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *SIGIR*, pages 250–257, 2001.
4. H. Cunningham. Information extraction - a user guide. Technical Report CS-97-02, University of Sheffield, 1997.
5. R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin, and D. P. Williamson. Searching the workplace web. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 366–375, New York, NY, USA, 2003. ACM Press.
6. D. Hawking. Challenges in enterprise search. In *15th. Australasian Database Conference*, 2004.
7. E. Kandogan, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu. Avatar semantic search: a database approach to information retrieval. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 790–792, New York, NY, USA, 2006. ACM Press.
8. I.-H. Kang. Transactional query identification in Web search. In *Asian Information Retrieval Symposium*, 2005.
9. I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *SIGIR*, pages 64–71, 2003.
10. I.-H. Kang and G. C. Kim. Integration of multiple evidences based on a query type for web search. *Information Processing Management*, 40(3):459–478, 2004.

11. W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 27–34, New York, NY, USA, 2002. ACM Press.
12. U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW05*, pages 391–400, 2005.
13. Y. Li, R. Krishnamurthy, S. Vaithyanathan, and H.V.Jagadish. Getting work done on the web: Supporting transactional queries. In *SIGIR*, 2006.
14. J. F. McCarthy and W. G. Lehnert. Using decision trees for coreference resolution. In *IJCAI*, pages 1050–1055, 1995.
15. K. Nanda. Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. In *Proc. of the 42nd Anniversary Meeting of the Association for Computational Linguistics (ACL04)*, 2004.
16. D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA, 2004. ACM Press.
17. A. Spink, D. Wolfram, M. B. J. Jansen, and T. Saracevic. Searching the web: the public and their queries. *J. Am. Soc. Inf. Sci. Technol.*, 52(3):226–234, 2001.
18. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2007. ACM Press.
19. T.S.Jayram, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu. Avatar information extraction system. *IEEE Data Engineering Bulletin*, May 2006.
20. T. Upstill, N. Craswell, and D. Hawking. Query-independent evidence in home page finding. *ACM Trans. Inf. Syst.*, 21(3):286–313, 2003.
21. H. Zhu, A. Löser, S. Raghavan, and S. Vaithyanathan. Navigating the intranet with high precision. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, 2007.