DynaQ — Faceted Search for Document Retrieval

Christian Reuschling, Stefan Agne, and Andreas Dengel
Knowledge Management Lab
German Research Center for Artificial Intelligence (DFKI)
Trippstadter Strasse 122, D-67663 Kaiserslautern, Germany
{christian.reuschling, stefan.agne, andreas.dengel}@dfki.de

ABSTRACT
'Sometimes it is easier to describe the way rather than the goal.' – this is especially true for searching documents in current desktop environments. Classical keyword search is only partially sufficient to manage large amounts of data, like the emails and documents of an average user. Several studies show that navigational approaches in searching fits better with the natural searching strategies of human beings.

This paper presents the DynaQ system aiming to increase information retrieval by combining faceted search with the state of the art searching paradigms *Orienteering* and *Dynamic Queries*. DynaQ stands for 'Dynamic Queries for document-based personal information spaces'.

Categories and Subject Descriptors
H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval—*Information filtering, Search process*

General Terms
Algorithms, Human Factors, Performance

Keywords
Document Retrieval Systems, Faceted Search, Desktop Search, Dynamic Queries, Orienteering

1. INTRODUCTION
Not only the amount of information available on the internet, but also the volume of information stored in personal desktop computers is constantly expanding and increasingly difficult to manage. Through the years thousands of emails and documents of all kinds create a complex personal information space. Naturally, this information space comprises many facets like title, author, and body, which should enable people to identify their information items, like documents.

In the domain of databases, due to the structured characteristic of the data, many facets are naturally available. This is not necessarily the truth in the document world. Facets like buzzwords, abstract, page count, etc. are optional and must be calculated, sometimes requiring a lot of effort.

In order to support facets to assist searching documents, search engines must consider these both in the backend and the user representation. There are many different ways to do this, and current applications already offer components such as index tree browsing. Nevertheless, there is a lack of work involving the heterogenious facets for search assistance, and normally the applications are not compatible with the natural search strategies of human beings.

DynaQ tries to go further in these directions, by combining faceted search technology with state of the art searching strategies, such as *Orienteering* [10] and *Dynamic Queries* [6].

Orienteering is typically characterized by relatively small steps following one after another. In contrast to searching through keywords, *Orienteering* offers several advantages: reduction of cognitive load, sensitivity of the environment, and a better understanding of the results [10].

Dynamic Queries enables the user to specify search queries in a dynamic way by getting immediate feedback on changes inside data sets that fulfill the query criteria [5].

2. STATE OF THE ART
The First Workshop on Faceted Search was held in conjunction with SIGIR 2006 in Seattle, WA. In the following you can find a quotation from the Call for Papers:

> Over the last few years, the direct search paradigm has gained dominance and the navigational approach became less and less popular. Recently a new approach has emerged, combining both paradigms, namely the faceted search approach. Faceted search enables users to navigate a multi-dimensional information space by combining text search with a progressive narrowing of choices in each dimension. It has become the prevailing user interaction mechanism in e-commerce sites and is being extended to deal with semi-structured data, continuous dimensions, and folksonomies....

There are several research groups in faceted search. Most of them focus on the user interface for realizing faceted search on structured data.

The *Relational Browser++* is a browsing tool for statistical information (developed at the University of North Carolina). 'It makes heavy use of mouse-overs to create a more dynamic and informative interface than is possible through most other kinds of interfaces' [7].

The mSpace browser is a multi faceted column based tool for browsing large data sets [9] (developed at the University of Southampton). This tool won the Semantic Web Challenge 2003.

Within the Flamenco search interface project under the lead of Marti Hearst at the University of California in Berkeley a framework is developed to organize the results of a keyword search by using hierarchical faceted metadata. FLAMENCO stands for FLexible information Access using MEtadata in Novel COmbinations [2].

Conclusion

Orienteering mirrors a typical strategy of humans when performing personally motivated searches, with relatively small steps following one after another. For this purpose, faceted search offers adequate support for the user. All the mentioned tools show that this approach is mainly realized for information spaces similar to data bases where data is clearly structured inside fixed, predefined attribute sets. As a rare exception Flamenco makes use of well structured meta data to browse the result list of a keyword search. The mission of DynaQ is to enable also the unstructured information parts of documents for the faceted search.

3. FACETS IN DYNAQ

In order to assist searching documents with facets, each information item must be represented in a structured, uniform data format. Thus, the available facets of a document become well formed, making them searchable. The heterogeneous nature of typical information items makes it impossible to search inside them directly because of lack of performance. Document formats have to be converted, and data is sometimes on the network and hard to access directly, e.g. email stored on an IMAP server. Additionally, it is impossible to assume a fixed data structure. DynaQ uses Lucene [3] to store documents as tuples of arbitrary attribute value pairs, highly optimized for search access.

To fill the storage, the search engine must deal with several data formats that the users work with, e.g. office formats (ODF, MS DOC) or emails. All documents have to be converted to the uniform data format, whereby all facets are extracted that are available directly inside a document. Typical examples for these are 'fullbody', 'title', 'author', 'pagecount', etc. DynaQ uses Aperture [1] for data extraction, which is able to convert a large range of common data formats into rdf containers. The content of these containers is stored inside a Lucene index by DynaQ.

Another category of facets are those who are not part of any document format, and thus must be calculated separately. Further, this category is split again into two facet types:

1. Calculated directly out of a single document, e.g. term stems, approximated page count (where needed), document language.

2. Calculated in relationship to all other user documents or other base corpora, e.g. document characteristic words (buzzwords), document similarities, auto-tagging, or classification. For this purpose, we get good results using indices which contain the contents of wikipedia [4].

To make sure that the same facets from different formats can be searched transparently, there has to be a common structure for a set of known attributes. One example is the body of a document which is called 'body' inside ODF and 'text' inside MS DOC. Both of these values have to appear under the same name inside the index because they are semantically identical. For this set of attributes, DynaQ makes use of the NIE ontology, specified by the Nepomuk consortium [8]. All other attributes will appear 'as is' inside the index and are still searchable under their original name.

4. FACETS INSIDE THE USER INTERFACE

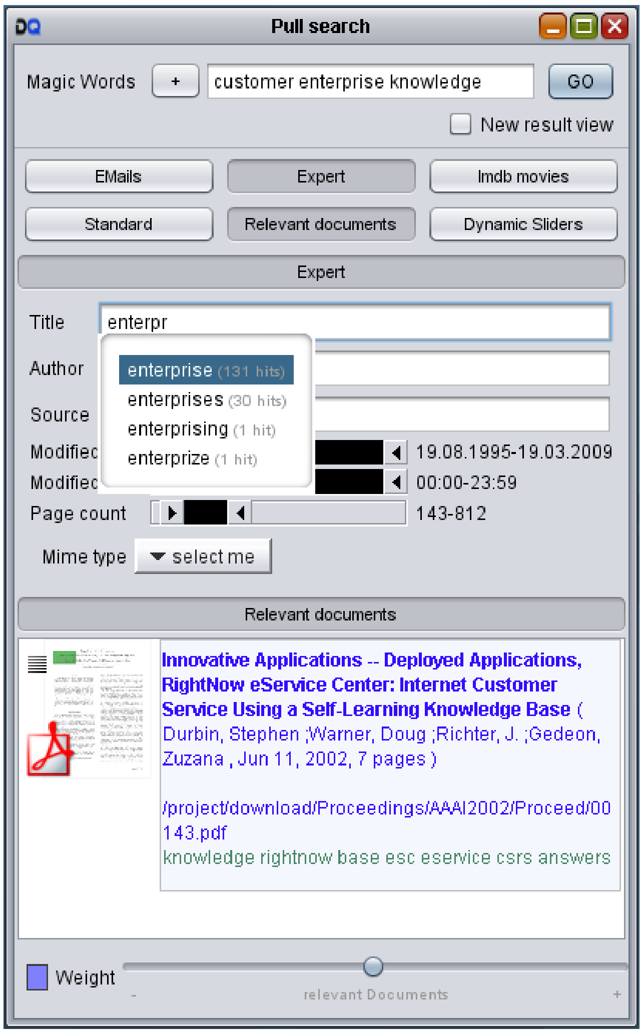
Once the facets are available in the backend for searching, they are still unusable to an average user. Thus, how an applications interface provides access to the facets is critical. This is a main focus of the DynaQ project. In fact, the range of usability of a frontend goes from 'totaly useless' to 'users find their data'.

Studies have shown that common searching enginges are not compatible with the natural search strategies of human beeings. Humans tend to search 'stepwise', while getting overview between each step, which is more a 'navigational' approach. DynaQ tries to support this searching paradigm, named *Orienteering* [10], by combining it with faceted search:

- Enabling *Dynamic Queries* approach [6] for searching facets gives 'stomach feeling' about the results.
- Setting context possibility with relevant documents (e.g. relevance feedback, 'document as facet'), document similarity search, and narrowing down on structured metadata (facets, e.g. date range).
- Explaining result documents with classical term highlighting and simple, automatic abstracting for getting document overview.
- Giving an overview, which most common result lists lack because of their focused view by realizing a birdeye view.

Dynamic Queries gives immediate feedback to the user by changing searching criterias, similar to the behaviour of a dimmer. This ad hoc response enables the user to get a feeling about how the result list changes by adjusting specific query dimensions / parameters. Traditionally, this approach is enabled in database-like environments, where the structure and the dimension are clear according to type and semantic.

For searching inside documents, DynaQ offers a mixed approach, whereby some database-like criterias can be adjusted directly. Thus, attributes with a well defined order can be used for shrinking the result list using range sliders. These are normally numerical ones, e.g. the creation date. Attributes with a set characteristic appear as checkbox menus for cherry-picking their values (e.g. document MimeType).



Magic Words: customer enterprise knowledge

Title: enterpr

enterprise (131 hits)
enterprises (30 hits)
enterprising (1 hit)
enterprize (1 hit)

19.08.1995-19.03.2009

00:00-23:59

Page count: 143-812

Innovative Applications -- Deployed Applications, RightNow eService Center: Internet Customer Service Using a Self-Learning Knowledge Base (Durbin, Stephen ;Warner, Doug ;Richter, J. ;Gedeon, Zuzana , Jun 11, 2002, 7 pages)

/project/download/Proceedings/AAAI2002/Proceed/00143.pdf

knowledge rightnow base esc eservice csrs answers

Figure 1: Query facets with contextualization

Inside unstructured data, it is natural to search for criteria defined by the user, normally in the form of query terms. This can also happen for semi-structured data, when several facets of a document comprise fulltext. For example, by searching emails, both the body and the subject are searchable. The same scenario is given inside attributes with a set characteristic, in the case the number of set values is too big for visualization (e.g. email author names). For these facets, DynaQ gives an edit field with type-ahead search, offering selectively the values for the current facet [Figure 1]. Spanning the dimensions for searching, each user query term appears as a 'term slider', enabling weighting of this specific dimension for searching. The documents inside the result list will be re-sorted immediately by adjusting the term slider [Figure 2].

Contextualization inside DynaQ takes two approaches. One approach is shrinking down the results by specifying dedicated metadata, as described. The other approach is a more fuzzy one, by marking documents as 'relevant'. For this, result documents can be picked and dropped inside a list of so called 'contextualization documents'. Thus, documents which are somehow similar to the contextualization documents appear higher inside the result list, where 'similar' means similar according to similar content / topics inside the documents. The similarity is computed statistically which brings a kind of fuzzyness and works with arbritrary documents. This has the benefit that all the possible user data can be considered. For the context consisting of documents marked as relevant, DynaQ also offers a weight slider which enables the Dynamic Queries approach for this synthetic facet [Figure 1]. To support orienteering, getting an overview is critical to know how to adjust your searching parameters adequately. To do so, there is a need for explanation components that describe why the documents inside the result list appear, i.e. shows the relationship of the documents to the users query. A common way to do so is query term highlighting, which is also realized in DynaQ. Each term (dimension) gets its own highlighting color for quick identification. As this approach adequately shows why the top documents appear, it does not give an overview of the whole result list. It is known that, in average, only the top N documents will be recognized by the user - all others are more or less 'hidden knowledge'. We have realized a bird's eye view of the result list, whereby the relevance of each query term is visualized per document, as a bar chart. For this, the highlighting colors assigned to the terms are used. This gives, for example, the chance to recognize that there still exists documents relevant to a specific query term that are not inside the top n documents. Further, in the case of document contextualization, you can see similarity peeks at context-relevant documents. The users are able to weight both term and context facets with the according sliders so that these documents will appear higher or lower inside the result lists immediately.

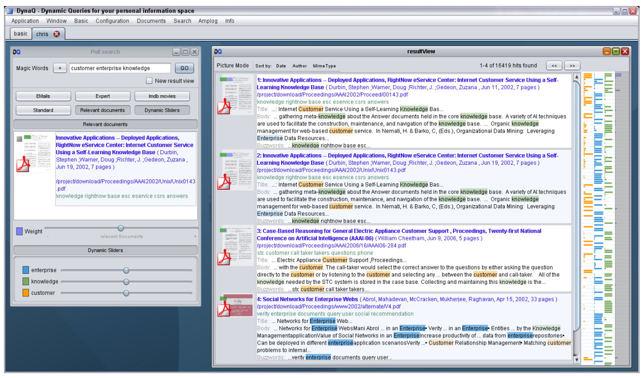


Figure 2: Query and results with birdeye view

Figure 2 gives an overview over a whole search, showing both the query window and the result list with the bird's eye view. The query weight sliders are at the bottom-left, the bird's eye view at the right, whereby the contextualization peeks are with grey background.

5. PERFORMANCE OF DYNAQ

The paradigms of DynaQ depend on the agility and performance of the application. The typical running environment is on the users desktop machine, thus it has to work on standard hardware.

A lot of effort was spent on runtime optimizations to realize this scenario. Here we want to show some examples of how DynaQ performs.

Environment: AMD Athlon(tm) 64 X2 Dual Core Processor 4400+, 2GB RAM

Two searchable indices, one with crawled desktop documents and emails (50 213 documents), combined with another one with the content of the german Wikipedia (1 315 192 documents) for parallel searching.

Average searching times (including visualisation):

- Two terms searched in fullbody and titles: ˜232 ms

  Number of result documents: 16 163

- With additional filtering over a facet (term inside the source attribute): ˜213 ms

  Number of result documents: 4 571

- With additional document contextualisation: ˜253 ms

  Number of result documents: 4 571

The time amount for the different types of queries is more or less the same. This is not surprising because most of the time is spent visualizing the results. A typical response time from the searching backend is about 25 ms, depending on the type of queries and the number of result documents that have to be ranked.

These times are sufficient to assure that dragging a weight slider gives the feeling that the result list moves smoothly.

6. SUMMARY

In this paper, we described the DynaQ system, a tool for searching the document based information space. It was shown how it deals with faceted search on heterogeneos data sources that are only partialy structured. In this context, the impacts of *Orienteering* and *Dynamic Queries* on both the backend and the frontend was described.

In contrast to the classical keyword search approach used in common search engines, DynaQ considers different kinds of facets, and implements methods for their use in information retrieval. Types of facets are

- Facets directly available inside the documents metadata (e.g. body, title, size, date, mime type).
- Facets that must be generated, derived from a single document (e.g. term stems, page count, language) and whole corpora (e.g. buzzwords, classification, auto tagging).
- Synthetic 'fuzzy facets' used for contextualization (document similarity).

The storage and access to the facet data was done very efficiently, which was critical to the implementation of the searching paradigmes. Thus, the DynaQ application is able to give agile feedback to the user while searching.

As open source software under the GPL2 license, DynaQ can be downloaded for common use at http://dynaq.opendfki.de.

7. ACKNOWLEDGMENTS

This research is supported by the *Stiftung Rheinland–Pfalz für Innovation.*

8. REFERENCES

[1] Aperture – a java framework for getting data and metadata. Available online: http://aperture.sourceforge.net/. Aperture – a Java framework for getting data and metadata.

[2] The flamenco search interface project. Available online: http://flamenco.berkeley.edu/. University of California, Berkeley.

[3] Jakarta lucene – a high-performance, full-featured text search engine library. Available online: jakarta.apache.org/lucene. Jakarta Lucene is a high-performance, full-featured text search engine library written entirely in Java.

[4] Wikipedia, the free encyclopedia. Available online: www.wikipedia.com. A Wiki-based encyclopaedia.

[5] S. Agne, C. Reuschling, and A. Dengel. DynaQ — Dynamic Queries for Electronic Document Management. In T. Kwok and W. Cheung, editors, *Proceedings of the First International EDM Workshop (EDM 2006) – The Electronic Document Management in an Enterprise Computing Environment*, pages 56–59, Hong Kong, China, 17. Oktober 2006. IEEE.

[6] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. In *Proc. ACM Conf. Human Factors in Computer Systems, CHI*, pages 619–626, 3–7 1992.

[7] R. G. Capra and G. Marchionini. The relation browser tool for faceted exploratory search. In *JCDL*, page 420, 2008.

[8] A. Mylka, L. Sauermann, M. Sintek, and L. van Elst. Nepomuk information element ontology. Available online: http://www.semanticdesktop.org/ontologies/nie/.

[9] M. Schraefel, M. Wilson, A. Russell, and D. A. Smith. mspace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM*, 49(4):47–49, 2006.

[10] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the 2004 conference on Human factors in computing systems*, volume 1, pages 415–422. ACM Press, 2004.