A Searchtask and Sub-searchtask Oriented Recommender

Hebein Gunther Cyrill

***Abstract*—** Most recommender systems work on data from one big single user session and rarely take into account different sessions for different goals. In real life, users split up their search into several sub-searchtasks for easier handling of their anomalous state of knowledge (ASK).

This poster presents an idea of an easy algorithm which takes this observation into account and could help improving the quality of recommending unseen information objects (IO) and, as a consequence, shortening the time it takes to normalize an anomalous state of knowledge is normalized and to answer a topical question.

This algorithm has not been fully implemented yet.

***Keywords*—**Information retrieval, Recommender, Relevance feedback, Sub-searchtask

I. Introduction

When searching for information on a given (wide spread) topic, the searching person usually faces an anomalous state of knowledge (ASK), i.e. a discrepancy between information need and available personal knowledge. As a consequence, this state leads to a craving for information to normalize this lack of knowledge [1], [2].

One way to solve the problem is to formulate searchtasks and to split these searchtasks further up into sub-searchtasks and so on, until such a sub-searchtask can be more or less easily answered or solved.

Hansen, Shepherd, Hackos and Redish define a task as "an activity to be performed in order to accomplish a goal." [3], where goal in this paper's sense means to find a sufficient answer to a given topical question or challenge.

A simple example of such a splitting up into sub-searchtasks can be found in Fig. 1:

Imagine someone who has bought a medieval house in Occitania (southern France) and would like to renovate this building (=goal/topic). Instead of executing one simple search on a request like "renovate old house", he introduces sub-searchtasks like "renovate roof" etc. These sub-searchtasks can have sub-searchtasks themselves, as shown with the searchtask "Renovate electricity", the sub-searchtask "220 Volts" and the sub-sub-searchtask "cables".

Each relevant information object found during research is not only relevant for the active searchtask itself but also for its super-searchtasks as well: An information object about "2,5 mm^2 three pole cable important for electric kitchen stove" would be relevant for the searchtask "cables", "220 Volts", "Renovate electricity" and the goal to renovate a medieval house, but it is less or not relevant for the searchtask "renovate roof".

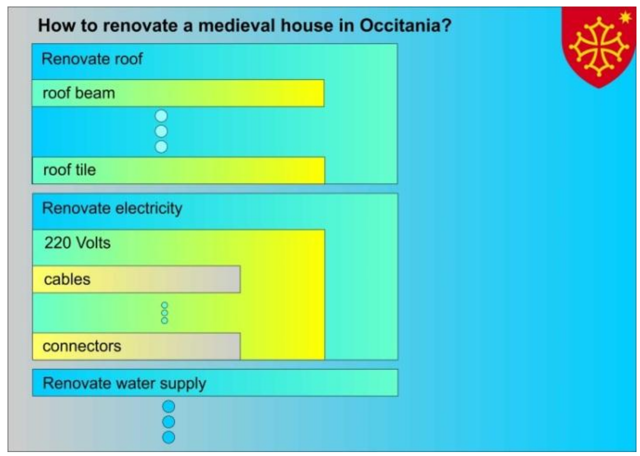

Fig. 1 Example of a topic search and its searchtasks

A. Search

Search algorithms usually rank results exclusively by comparing a given request to representations of information objects, but do not take into account contextual information ("awareness"). So gaining optimized knowledge in the big and growing universe of information is getting harder and sometimes feels nearly impossible using pure search algorithms [4]–[6].

In many areas, such search algorithms have been replaced or extended by recommender systems. These take into account contextual information, like for example user profiles, different interests, different times, geographical data etc.

B. Recommendation

Such systems are based on recommendation algorithms that can be classified into three types:

1. *Collaborative recommendation* (Collaborative Filtering, CF) algorithms are mostly based on (anonymized) user profiles ("who buys what?") and try to find relevant objects for a distinct user by comparing one user's profile to other users' profiles.

Hebein Gunther Cyrill is Austrian pharmacist and PhD-student of computer sciences at the Department of Multimedia and Internet applications of the University of Hagen, Germany (e-mail: gunther.hebein<[at>]fernuni-hagen.de).

This type of recommendation assumes that different users have similar interests and that these users can be grouped.

A profile with at least 15 objects is needed to calculate useful results. As 15 objects per user is quite a large precondition, collaborative recommendation is not the algorithm of first choice for topical task-based information search.

2. *Content based recommendation* algorithms compare information objects, especially their content or meta-data, to a reference profile. As content is needed for calculation, quality and quantity of data and their chosen representation have direct influence on the results.

3. *Knowledge based recommendation* algorithms are used in very rare cases (like diagnostic recommendations in medicine or pharmacy) and are based on experts' knowledge [4]–[7].

Depending on the type and amount of data available, those three types are used in different scenarios [6], [8], [9].

[10] shows a table to choose the right type of algorithm for a given problem by comparing several parameters.

C. Relevance Feedback

As search and recommendation are calculated by an algorithm, the system needs some kind of feedback by its user(s). This feedback is based on the relevance of an information object.

The Oxford Dictionary defines **relevance** in a very abstract manner as "property of fulfilling the requirements of a user's search for information" and "the degree to which a document [...] fulfills such requirements." [11]

Beneath its many faceted appearances, relevance neither seems to be totally comprehensible nor to be quantifiable, only comparable [12], [13] and as a consequence relevance has been one of the main challenges in information retrieval since its very beginning [14], [15].

Statements on relevance either can be explicit (someone/something says directly, that an object is relevant) or can be expressed implicitly.

Explicit feedback must be expressed in an active manner, whereas **implicit feedback** is coded in nearly every interaction between human user and an information object (GUI/HID-interaction).

II. IDEA

Our idea is to use this hierarchical treelike structure of searchtasks and sub-searchtask by attaching information objects to it and by recommending potent interesting information objects for each of its nodes and subtrees. The tree's structure itself is not known from the beginning of the search, but develops over time during the information retrieval process.

A. Exploring the Searchtask Tree's Structure

Knowing the structure of the searchtask-tree for a distinct topic or goal and knowing about the context allows the user to attach navigated and inspected information objects to a defined sub-searchtask. A group of attached information objects can be further-more processed to get recommendations based on them (see Fig. 3 Example of a searchtask and sub-searchtask tree structure).

Information objects that have been recommended this way can refer to the searchtask, the super-searchtask and the topic itself.

To reveal the treelike searchtaskstructure two possibilities generally exist:

1. The **user himself** expresses what searchtasks he would like to create and to solve: Fig. 2 shows a screenshot from "Tasks to do". It is an early stage addon for Firefox, where users can define searchtasks and sub-searchtasks and are able to attach information objects directly to such a node.

2. **Analyzing user behavior** and grouping similar requests into one subtask allows for the calculation of possible trees as well. These are only estimates and may differ from the user's point of view.

Both processes of exploring the structure are continuous. Sub-searchtasks can appear at any time during an active search.

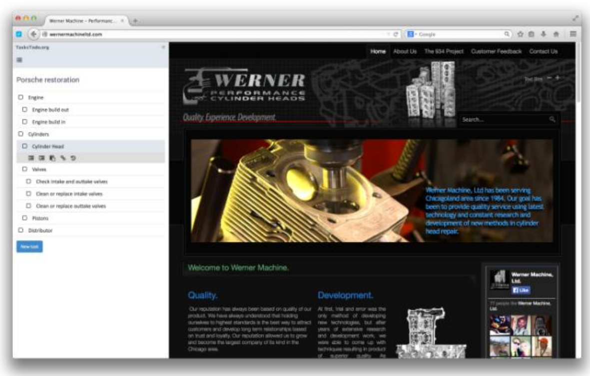

Fig. 2 Screenshot from Backhausen's "Tasks to do"

B. Adding information objects to sub-searchtasks

As soon as a new searchtask has been identified, information objects can be added manually or automatically:

1. The **manual (=explicit) way** allows the user to decide for himself if an information object is contextually relevant for the actual searchtask or maybe another, not active tasks. The user himself indicates where to put the information object in the tree.

2. The **automatic (=implicit) way** watches the user and his interactions with an information object. The system decides whether this interaction may be a hint for relevance and in some cases how big this relevance might be (relatively, not absolutely) [5], [7]. The information object then can be automatically added to the active sub-searchtask or, if no active one is declared, the system calculates which sub-searchtask may be the active one at the moment of interaction.

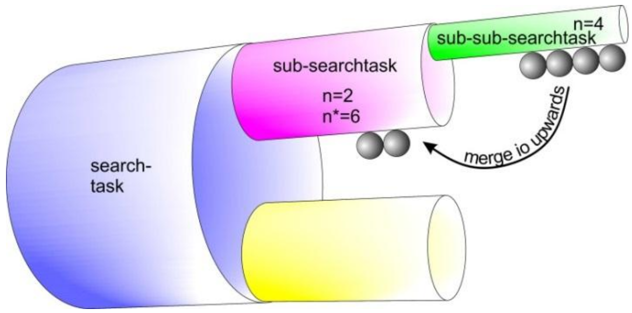

Fig. 3 Example of a searchtask and sub-searchtask tree structure

C. Recommendation Tool

From a searchtask and sub-searchtask tree with attached information objects, recommendations for several parts of that tree may be calculated. These calculations can be accomplished on content- or on collaboration-based recommendation, depending for example on

- the type of information objects
- information included in information object
- the homogeneity of information objects
- the risk of wrong recommendation
- the type of interaction between user and algorithm
- the stability of users preferences
- the transparency of recommendations
- the stability of information in the information universe
- the total number of information objects in the node
- the total number of available information objects in the universe
- the total number of collaborative data from other users [10].

The simplified algorithm works as follows:

1. Execution of **recommendation**
2. The user requests recommendation for a distinct (sub) searchtask.
3. Optional: If a new information object is added to a (sub-)searchtask, a recommendation is started automatically.
4. For this (sub-)searchtask **load** all **user-relevant** information **objects**
5. If there are fewer information objects available than needed, recursively merge all information objects from sub-searchtasks, until the minimal number of information objects is met.
6. **Create** a useful **representation** of all documents for the searchtask, which takes into account all attached documents.
7. **Use the search engine** with this representation (at the moment Google API, implementation as module to be able to use other APIs as well)
8. From the retrieved ranked hit list, **select top information objects**, which
    a. are not known as relevant by now
    b. have not been presented/seen yet (with a certain possibility to be presented though already seen)
    c. have not been marked as not relevant yet.
9. **Present** this selection to the user in a user friendly way (not aggressive or disturbing), like shown in Fig. **4**.
10. The user himself may tag a new information object as relevant, irrelevant or relevant for another searchtask.

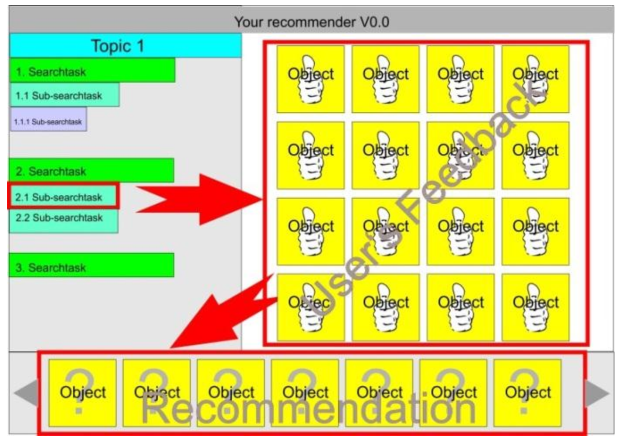

Fig. 4 Sample for graphical user interface

III. Data Sources

To show whether splitting a search into searchtasks and sub-searchtasks could be useful, a data collection is needed.

As tools like "Tasks to do" do not have a big pool of users and data as of now, other data sources ought to be used.

One data collection which has a similar layout with is the sessiontrack-collection from the TREC [16]–[19]:

Different people were asked to answer questions on distinct topics. Their searches were session based and had to be documented in every detail (searchtask, sub-searchtasks, purpose of the request, and quality of the information objects).

This data collection is freely available (years 2011, 2012, and 2013) and only has to be transferred from XML to a relational format (SQL) in order to be easily processable by the prototype of our recommender module.


```
<sessiontrack2011>
   <session num="1" starttime="08:59:47.258675">
      <topic>
         <title>peacecorp</title>
         <desc>Find information about the peace corp</desc>
         <narr>When was it started and by whom? What services does it
provide and where does it provide these services? What is the criteria for
applying? What is the salary or stipend? What positions are available?</narr>
      </topic>
      <interaction num="1" starttime="09:00:04.155323">
         <query>peace corp</query>
         <results>
            <result rank="1">
               <url>http://www.peacecorps.gov/</url>
               <clueweb09id>clueweb09-en0011-60-08003</clueweb09id>
               <title>Peace Corps</title>
               <snippet>Fighting hunger, disease, poverty, and lack of
opportunity.</snippet>
            </result>
            <result rank="2">
               <url>http://en.wikipedia.org/wiki/Peace_Corps</url>
```

```
                    <clueweb09id>clueweb09-enwp01-43-22314</clueweb09id>
                    <title>Peace Corps - Wikipedia, the free encyclopedia</title>
                    <snippet>The Peace Corps is an American volunteer program
run by the United States Government, as ... The mission of the Peace Corps
includes three goals: providing technical assistance, ...</snippet>
                </result>

            </results>
            <clicked>
                <click num="1" starttime="09:00:09.943356"
endtime="09:01:13.434255">
                    <rank>1</rank>
                </click>
                <click num="2" starttime="09:01:18.582078"
endtime="09:02:42.552354">
                    <rank>2</rank>
                </click>
[...]
            </clicked>
        </interaction>
        <interaction num="2" starttime="09:02:55.569644">
            <query>peace corp apply</query>
            <results>
                <result rank="1">
                    <url>http://www.peacecorps.gov/</url>
                    <clueweb09id>clueweb09-en0011-60-08003</clueweb09id>
                    <title>Peace Corps</title>
                    <snippet>Peace Corps Volunteers travel overseas to make real
differences in the lives of real people. Apply online to Volunteer, find a local
recruiting event, donate to a ...</snippet>
                </result>
[...]
                </results>
            <clicked>
                <click num="1" starttime="09:03:02.615239"
endtime="09:03:33.507677">
                    <rank>2</rank>
                </click>
[...]
            </clicked>
        </interaction>
        <currentquery starttime="09:04:03.469341">
            <query>peace corp application</query>
        </currentquery>
    </session>
<sessiontrack2011>
```


Fig. 5 Sample data from TREC - Sessiontrack 2011 [17]

IV. Research Questions

The main questions of this paper are to find out

- whether it is possible to calculate recommendations for searchtasks and separately for their sub-searchtasks,
- if results for a sub-searchta,sk are relevant to its super-searchtask
- if splitting a topical search into searchtask and sub-searchtasks could lead to better and faster satisfaction of the user and his anomalous state of knowledge,
- if with sparse information objects, sparsely filled subtasks can be merged up into the super-searchtask and
- what the minimal number of information objects is to get a useful recommendation.

V. Acknowledgments:

The author wants to thank Prof. Mathias Hemmje and Dr. Paul Landwich (University of Hagen, Germany, Department of Multimedia and Internet applications), and Dr. Claus-Peter Klas (GESIS - Leibniz Institute for the Social Sciences Cologne, Germany) for their support.

References

[1] N. J. Belkin, R. N. Oddy, and H. M. Brooks, "ASK for information retrieval: Part I. Background and theory," J. Doc., vol. 38, no. 2, pp. 61–71, 1982. http://dx.doi.org/10.1108/eb026722

[2] N. J. Belkin, R. N. Oddy, and H. M. Brooks, "ASK for information retrieval: Part II. Results of a design study," J. Doc., vol. 38, no. 3, pp. 145–164, 1982. http://dx.doi.org/10.1108/eb026726

[3] P. Vakkari, "Task-based information searching," Annu. Rev. Inf. Sci. Technol., vol. 37, no. 1, pp. 413–464, Jan. 2003. http://dx.doi.org/10.1002/aris.1440370110

[4] S. Büttcher, C. L. A. Clarke, and G. V. Cormack, Information Retrieval: Implementing and Evaluating Search Engines. Mit Pr, 2010.

[5] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, 2ed edition. Addison Wesley, 2010.

[6] D. Jannach, M. Zanker, A. Felfernig, and G. Friedich, Recommender Systems: An Introduction. Cambridge University Press, 2010. http://dx.doi.org/10.1017/CBO9780511763113

[7] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, pp. 1–35. http://dx.doi.org/10.1007/978-0-387-85820-3

[8] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," Recomm. Syst. Handb., pp. 217–253, 2011.

[9] R. D. Burke, A. Felfernig, and M. H. Gker, "Recommender systems: An overview," AI Mag., vol. 32, no. 3, pp. 13–18, 2011.

[10] R. Burke and M. Ramezani, "Matching Recommendation Technologies and Domains," Recomm. Syst. Handb., pp. 367–386, 2011.

[11] Relevance," Oxford Dictionary Online. 10-Sep-2014.

[12] A. M. Rees and T. Saracevic, The measurability of relevance. Center for Documentation & Communication Research, Western Reserve University, 1966.

[13] S. Mizzaro, "Relevance: The whole history," J. Am. Soc. Inf. Sci., vol. 48, no. 9, pp. 810–832, Sep. 1997. http://dx.doi.org/10.1002/(SICI)1097-4571(199709)48:9<810::AID-ASI6>3.0.CO;2-U

[14] E. Cosijn and P. Ingwersen, "Dimensions of relevance," Inf. Process. Manag., vol. 36, no. 4, pp. 533–550, Jul. 2000. http://dx.doi.org/10.1016/S0306-4573(99)00072-2

[15] L. Schamber, M. B. Eisenberg, and M. S. Nilan, "A re-examination of relevance: toward a dynamic, situational definition∗," Inf. Process. Manag., vol. 26, no. 6, pp. 755–776, 1990. http://dx.doi.org/10.1016/0306-4573(90)90050-C

[16] "Text REtrieval Conference (TREC) Home Page." [Online]. Available: http://trec.nist.gov/. [Accessed: 10-Sep-2014].

[17] "Text REtrieval Conference (TREC) 2011 Session Track." [Online]. Available: http://trec.nist.gov/data/session2011.html. [Accessed: 10-Sep-2014].

[18] "Text REtrieval Conference (TREC) 2012 Session Track." [Online]. Available: http://trec.nist.gov/data/session2012.html. [Accessed: 10-Sep-2014].

[19] "Text REtrieval Conference (TREC) 2013 Session Track." [Online]. Available: http://trec.nist.gov/data/session2013.html. [Accessed: 10-Sep-2014]