Webis at the CLEF 2017 Dynamic Search Lab

Matthias Hagen, Johannes Kiesel, Milad Alshomary, and Benno Stein

Bauhaus-Universität Weimar
<first name>.<last name>@uni-weimar.de

Abstract We briefly describe our approach to the query suggestion task at the CLEF 2017 Dynamic Search Lab. The general research idea of our contribution is to evaluate query suggestions in form of keyqueries for clicked documents. A keyquery for a document set D is a query that returns the documents from D among the top-k ranks. Our query suggestion approach derives keyqueries for pairs of documents previously clicked by the user. The assumption then is that the not-already-clicked documents in the top results of the keyqueries could also be interesting to the user. The keyquery suggestions thus focus on retrieving more documents similar to the ones already clicked. Another reasonable approach might instead focus on suggesting queries covering aspects of a user's information need different to the ones in the already seen documents. However, in our this year's contribution to the Dynamic Search Lab, we explore the utility of the "similar" suggestions that keyqueries probably produce.

Approach

A query q is a keyquery for a document set D iff q returns the documents from D in its top-k ranks, q has at least l results, and no subquery of q has the previous two properties [4]. The parameters k and l describe the specificity and generality of a keyquery (typically k would be small and l be in the range of 10 or a 100) while the last property ensures minimality in a set theoretic sense. We have succesfully applied keyqueries to generate dynamic taxonomies in the context of digital libraries [5,10], to identify similar web pages [7], to support scholarly search for related work [6], and for document clustering + labeling [3].

The research question addressed in our Dynamic Search Lab contribution is that of how useful query suggestions in form of keyqueries for clicked documents are. We conjecture that such queries represent different formulations of information needs very related to the one that made the user click.

The Dynamic Search Lab data contains 26 topics [8], along with one query submitted by some user in a search session and the shown results from the whole session with some indicated as being clicked by the user. Exactly for these clicked documents, we derive keyqueries as query suggestions. For topic i, we identify the documents D_i that were clicked from the result lists in the session and that are contained in the ClueWeb12 Part B (the collection behind the Dynamic Search API) and that have a spam ranking of at least 30 [2] (i.e., at least 30% of

the ClueWeb12 are more spammy). Thus, D_i consists of documents that can be assumed to be somewhat relevant to q_i, are in the document collection of the Dynamic Search API, and are not the most spammy.

For the documents in every D_i we try to derive five keyqueries as query suggestions. We first extract the main content from the documents in D_i by using all sentences with at least one English stop word from text paragraphs of at least 400 characters [9]. The clicked documents are arranged in pairs (first and second clicked document, third and fourth clicked document, etc.). For each such pair, the contents are concatenated and the top-10 keyphrases (not longer than three words) are extracted using the head noun extractor [1]. We also filter out keyphrases that are too "specific," namely returning fewer than 100 results against the Dynamic Search API. For the extracted keyphrases of the first pair of clicked documents, we derive a keyquery cover [6] setting $k = 50$ (i.e., the two clicked documents are returned in the top-50 results) and $l = 100$ (i.e., the query needs to return at least 100 results).

If the keyquery cover of the first pair already contains five keyqueries, we stop. Otherwise, a keyquery cover for the second pair is computed etc. Using this approach, we generate a total of 66 query suggestions for 19 topics. For the remaining seven topics, one does not have any clicks and for six topics no keyquery cover could be computed for any pair of clicked documents.

As for the ranked list for a topic, we use the top-10 results of each of the at most five derived keyqueries returned by the Dynamic Search API and merge them as follows: first the first ranks of the queries, then the second ranks, etc.; duplicate results that already are in the merged list are replaced by the next result from the same keyquery.

References

1. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. AI 2000, pp. 40–52.
2. Cormack, G., Smucker, M., Clarke, C.: Efficient and effective spam filtering and re-ranking for large web datasets. Information Retrieval 14(5), 441–465 (2011)
3. Gollub, T., Busse, M., Stein, B., Hagen, M.: Keyqueries for clustering and labeling. AIRS 2016, pp. 42–55.
4. Gollub, T., Hagen, M., Michel, M., Stein, B.: From keywords to keyqueries: Content descriptors for the web. SIGIR 2013, pp. 981–984.
5. Gollub, T., Völske, M., Hagen, M., Stein, B.: Dynamic taxonomy composition via keyqueries. JCDL 2014, pp. 39–48.
6. Hagen, M., Beyer, A., Gollub, T., Komlossy, K., Stein, B.: Supporting scholarly search with keyqueries. ECIR 2016, pp. 507–520
7. Hagen, M., Glimm, C.: Supporting more-like-this information needs: Finding similar web content in different scenarios. CLEF 2014, pp. 50–61.
8. Kanoulas, E., Azzopardi, L.: Overview of the CLEF Dynamic Search Evaluation Lab 2017. CLEF 2017.
9. Kiesel, J., Stein, B., Lucks, S.: A large-scale analysis of the mnemonic password advice. NDSS 2017.
10. Völske, M., Gollub, T., Hagen, M., Stein, B.: A keyquery-based classification system for CORE. WOSP 2014.