Leveraging Wikipedia's article structure to build search agents

TUW at CLEF 2017 Dynamic Search

Joao Palotti

Vienna University of Technology (TUW)
Favoritenstrasse 9-11/188 1040
Vienna, Austria
palotti@ifs.tuwien.ac.at

1 Introduction

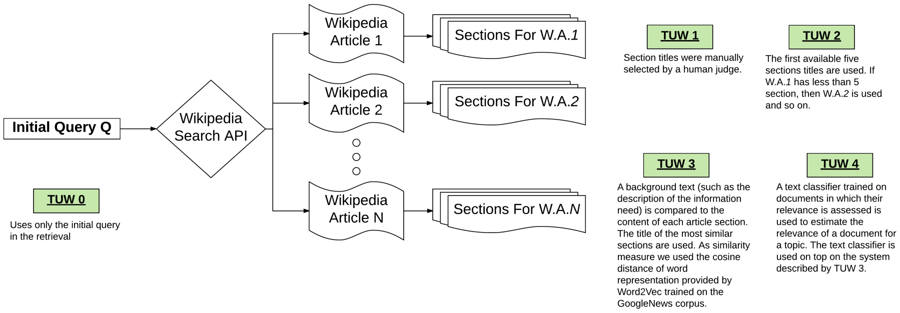
Often, single query search sessions are not enough to solve complex problems or to gather sufficient information to take an informed decision. Such complex search tasks include many ordinary tasks such as planning a vacation trip, studying for a school or college exam or gathering information on a symptom or condition. Nevertheless, complex search tasks can be broken into multiple smaller specific subtasks. In order to assist users in dealing with complex searches, a *search agent* could be employed to automatically break a complex search task into smaller tasks, to issue multiple queries for those subtasks, and to report the results back to the user in a meaningful way.

A key problem that the Information Retrieval community aims to solve in order to create such agents is the understanding of complex search tasks, which includes the identification of smaller subtasks. To foster research in such interesting problem a number of challenges have been recently proposed (e.g., [5,4,2]) and this paper describes the efforts of Vienna University of Technology (TUW) in one of such challenges, the first CLEF Dynamic Search [2].

We propose the creation of a search agent that specifically leverage the structure of Wikipedia articles to understand search tasks. Our assumption is that human editors carefully choose meaningful section titles to cover the various aspects of an article. Our proposed search agent explores this fact, being responsible for two tasks: (1) identifying the key Wikipedia articles related to a complex search task, and (2) selecting section titles from those articles.

For instance, consider a user seeking information on how to quit smoking. Some of the relevant subtasks, in this case, are the description of different ways to quit smoking, the benefits of quitting smoking and second effects of quitting smoking. A possible query that expresses this information need is simply *"quit smoking"*. The Wikipedia article *Smoking Cessation*[1] is the top hit for such query

[1] https://en.wikipedia.org/wiki/Smoking_cessation



TUW 0: Uses only the initial query in the retrieval

TUW 1: Section titles were manually selected by a human judge.

TUW 2: The first available five sections titles are used. If W.A.*1* has less than 5 section, then W.A.*2* is used and so on.

TUW 3: A background text (such as the description of the information need) is compared to the content of each article section. The title of the most similar sections are used. As similarity measure we used the cosine distance of word representation provided by Word2Vec trained on the GoogleNews corpus.

TUW 4: A text classifier trained on documents in which their relevance is assessed is used to estimate the relevance of a document for a topic. The text classifier is used on top on the system described by TUW 3.

Fig. 1: Five runs were submitted at CLEF Dynamic Search: one simple baseline made up using only the initial query and four runs made appending section names extracted from Wikipedia to the initial query.

using the Wikipedia Search API[2], and many of the crucial aspects of this topic are presented in the various sections of this article, e.g. *methods*, *side effects* and *health benefits*. Our proposed agent benefits from the effort made by the human editors on Wikipedia to easily gather information on all these aspects. One feature of our method is its accountability as it is easy to explicitly justify to users which are the subtasks being considered.

In the next section, we describe the details of our approach, including the methods devised to rank section titles from Wikipedia articles. In Section 4 we discuss our results and future work directions.

2 Experiments

Our proposed search agent starts by redirecting an initial user query to the Wikipedia API, which retrieves the top N Wikipedia articles for that query. We download the full-text of all top N Wikipedia articles retrieved by the Wikipedia API (which was accessed in March 2017), and we evaluate different approaches to select up to 5 section titles from the downloaded articles. Each section title is considered a subtask relevant for one aspect of the original user information need. We then append the selected section titles to the initial user query and issue multiple queries to an ElasticSearch index of ClueWeb 12 Category B, provided by the organizers. We do not prioritize any subtask and merge the results from the different subtasks using a round-robin approach. Other ways to present the results will be explored in future work. In this work, we set $N = 3$ and access the Wikipedia API using the Python package Wikipedia version 1.4.0[3].

[2] Accessed on 25th May 2017:
`https://en.wikipedia.org/w/api.php?action=query&list=search&srsearch=quit%20smoking&utf8=`

[3] `https://pypi.python.org/pypi/wikipedia/`

In total, we submitted five runs: one baseline run (TUW 0) and four regular runs (TUW 1-4) as following described:

- **TUW0 – Baseline Run:** This is an ElasticSearch BM25 run simply retrieving 50 documents using the query field for each information need. This run aims to assess how the usage of a search agent improves or hurts the search results;
- **TUW1 – Human Run:** A human judge is used to select up to 5 section titles from the top Wikipedia articles returned by the Wikipedia API. This run aims to provide a comparison between automatic and manual methods to choose section titles;
- **TUW2 – First_Five Run:** This run implements a simple heuristic in which the first five section titles are selected. The assumption made here is that the Wikipedia Search API is highly precise, thus the top articles are more relevant than the others. Also, we assume that the most important aspects of a topic are addressed early in a Wikipedia page.
- **TUW3 – Word2VecMean Run:** In this run, we take advantage of a background text to automatically rank Wikipedia section titles based on the content of each section. As background text, we used the description of the information need provided by the Workshop organizers and compare it to the text used in each individual section. We make use of the cosine similarity between the vector representation of each word in the both texts. The vector representations are extracted using Word2Vec trained on GoogleNews [3]. Given the operator # that returns the number of elements of a set, we score each section S comparing the Word2Vec representation (W2VR) of each word WS in the section with each word WD in the information need description D. Formally:

$$Score(S) = \frac{\sum_{SW \in S} \sum_{DW \in D} Cosine(W2VR(SW), W2VR(DW))}{\#S \times \#D}$$

- **TUW4 – W2V_Plus_NB Run:** This run adds a step to TUW3. After selecting the top section titles and before applying the round-robin algorithm to merge the result of each query, an automatic text classifier predicts the relevance of each retrieved document. We used a Naive Bayes classifier trained on a set of documents that were judged with respect to their relevance to the topic. The reason to choose a Naive Bayes classifier is that it is a standard approach for text classification. Other text classifiers will be evaluated in future work.

An overview of our method and runs representing different strategies to select the top 5 section titles from the top Wikipedia articles is depicted in Figure 1.

3 Results

In Figure 2, we show different standard evaluation metrics and their results for each run. Figure 3 shows the distribution of Precision at depth 5 and 10 over topics.

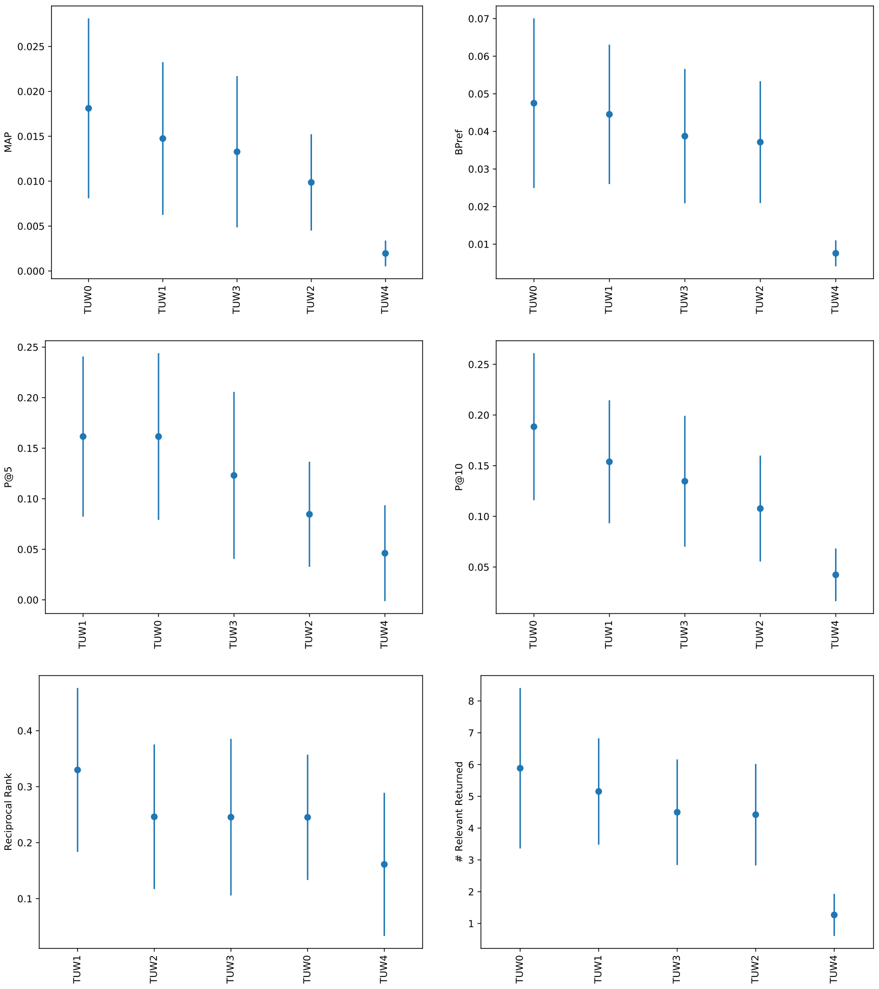


Fig. 2: Mean Average Precision (MAP), Binary Preference (BPref), Precision at depth 5 (P@5), Precision at depth 10 (P@10), Reciprocal Rank and the average number of relevant documents returned are plotted for each run. Runs are sorted by their average results over all topics and the error bar represents a confidence interval of 0.95 around the mean.

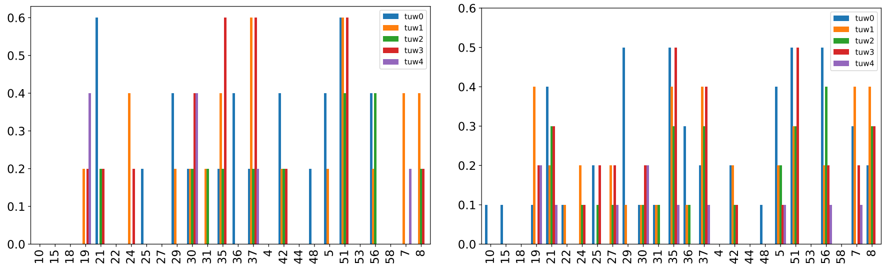


Fig. 3: Distribution of Precision at depth 5 (left) and Precision at depth 10 (right) over all the topics for each submitted run.

4 Discussion and Conclusion

Overall, our baseline run, TUW0, obtained the best results, although the difference between TUW0 and TUW1-3 is not statistically significant for any metric shown in Figure 2. In some cases, as for P@5 and Reciprocal Rank, the human selection of titles from Wikipedia article was on average better than the baseline.

Interestingly, Figure 3 shows that Topic 4 (quit smoking) had no relevant documents for any of our runs. An analysis of the QRels and retrieved pages might be necessary, as, for example, we did not remove any spam webpages and this topic is potentially vulnerable to spam. Considering P@5, the results using any search agent outperformed the baseline in 6 topics (7, 8, 19, 24, 27 and 31), while the baseline was exclusively better than any search agent in 7 topics (5, 21, 25, 29, 36, 42 and 48).

Intent aware metrics, such as α-NDCG or ERR-IA, were not used in this evaluation, however they could reveal the potential benefits of using our proposed search agent. Experiments with this kind of metrics is left as future work.

References

1. Darío Garigliotti and Krisztian Balog. The university of stavanger at the TREC 2016 tasks track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.
2. Evangelos Kanoulas and Leif Azzopardi. Overview of the clef dynamic search evaluation lab 2017. In *CLEF 2017 - 8th Conference and Labs of the Evaluation Forum.* Lecture Notes in Computer Science (LNCS), Springer, 2017.
3. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
4. Manisha Verma, Emine Yilmaz, Rishabh Mehrotra, Evangelos Kanoulas, Ben Carterette, Nick Craswell, and Peter Bailey. Overview of the TREC tasks track 2016. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.

5. Emine Yilmaz, Manisha Verma, Rishabh Mehrotra, Evangelos Kanoulas, Ben Carterette, and Nick Craswell. Overview of the TREC 2015 tasks track. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC, Gaithersburg, Maryland, USA*, 2015.