Context Sensitive Search Engine

Remzi Düzağaç and Olcay Taner Yıldız

Abstract In this paper, we use context information extracted from the documents in the collection to improve the performance of the search engine. In first step, we extract context using Lucene, DBPedia-Spotlight, and Wordnet. As the second step, we build a graph using extracted context information. In the third step, in order to group similar contexts, we cluster context graph. In the fourth step, we re-score results using context-clusters and context-information of documents, as well as queries. In the fifth step, we implement a data collection tool to collect gold-standard data. In the sixth and final step, we compare the results of our algorithm with gold-standard data set. According to the experimental results, using context information may improve the search engine performance but the collection should be relatively big.

1 Introduction

Search engines deal with the measurement of how close the source information matches with the user input; thus retrieving the most relevant information to the users. In order to calculate the relevancy, search engines use several parameters such as the popularity of a document, the date of a document, user preferences extracted from usage logs, etc. Therefore, the search engines generally are not able to consider the *meaning* of the input, in other words, it cannot differentiate the context that the search input is related to. This is why the user sometimes might see search results that seem completely out of context comparing to what they searched for. This important problem is the main focus of this paper.

The capability of processing information of the human brain may provide a viable basis for new approaches. In order to process large amount of data, the human brain generates relationships between the current sensory information that represents the external world and previously stored information. This relationship allows human

R. Düzağaç (✉) · O.T. Yıldız (✉)
Işık University, Sile, Istanbul
e-mail: remzi.d@gmail.com

O.T. Yıldız
e-mail: olcaytaner@isikun.edu.tr

© Springer International Publishing Switzerland 2014

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_29

olcavtaner@isikun.edu.tr

brain to determine actions. This process in the brain provides us an inspiring idea about how the search engines might be improved by adopting the "context over content" approach [11]. This is what this paper contributes to the field of search engines: adding context- based search capability (retrieving documents that do not only contain same words with the search query but also belonging to similar contexts). Instead of focusing on finding the exact content, we take the context from the input information into account and improve the relevancy rate of the results and thus the overall performance of the search.

This paper is organized as follows: We give related work in Sect. 2 and describe the proposed system in Sect. 3. We explain the experimental setup and give results in Sect. 4. Finally, we conclude in Sect. 5.

2 Related Work

In the study of Brown et al. [4], Context Aware Retrieval (CAR) systems and its relationships to information retrieval and filtering are surveyed. They claim that CAR is a sparsely researched area and there are some topics of information retrieval and filtering which is not applied to CAR systems.

Mylonas et al. [13] bring two fields of information retrieval together to present a reliable knowledge-based information extraction method: personalization of the information and retrieval context. The authors state that, this combination provides a unique approach towards the process of measuring relevance, since it introduces the concept of ontological information domain which refers to the enriched representational ground for content meaning, user interests, and contextual conditions.

Li et al. [10] make use of semantic distance for feature-free search query classification. They show advantages of utilizing the number of search results while grouping the web content according to their relevance. In contrast to other machine learning (ML)-based methods, they use a feature-free approach since it's a better fit for ever-changing web data.

In the study of [7], Concept-Based Information Retrieval is performed using explicit semantic analysis. Their concept-based approach (which seems similar to our context over content approach) involves three important contributions: using Explicit Semantic Analysis which refers to real-life concepts resembling human perception, integrating selection phase into concept-based indexing stage, and using three AI-based selection methods for information retrieval.

Atanas Kiryakov et al. [8] introduce an architecture for semantic annotation, indexing and retrieval of documents via semantic resources. Their architecture claims to provide automatic semantic annotation with references to ontology classes and instances. They perform semantic indexing and retrieval where they combine ontology and information retrieval methods.

Schuhmacher and Ponzetto [14] state a knowledge-based approach to web search result clustering and try to solve related research topics focusing on clustering short texts from the web. This information retrieval task is analyzed if it can take advantage

olcavtaner@isikun.edu.tr

from DBpedia spotlight state-of-the-art entity linking system. The approach uses DBpedia concepts as text seeds to collect topical concept labels. These labels are used to cluster based on their topical similarity. The approach takes web search snippets as inputs and cluster them topically using DBpedia.

3 Proposed System

The system that we designed to implement the context over content approach consists of five different steps: corpus, context extraction, context clustering, labeling, and scoring (Fig. 1). The first step is corpus, which contains preprocessed documents and queries for next steps. The second step is the context extraction. In this part, we use Lucene, DBPedia Spotlight, and Wordnet to extract context information from documents and queries. The third step is the context clustering. We cluster extracted contexts to form groups, which contain similar contexts.Fourth step is labeling. In the labeling step, we use context information extracted from each document and clusters to determine the clusters that each document is related to. Also we label queries in the same manner with documents. Fifth and final step is scoring. In the scoring step, we calculate scores according to the correlations between queries and documents. Brief description of each step is given in the paper. For more detail, see [6].

3.1 Corpus

A corpus is a collection of texts in digital form, over which processes such as retrieval or analysis will performs [5]. In this paper, we use The Westbury Lab Wikipedia Corpus (WLC) [15], as our data source. We apply some preprocessing steps to WLC in order to obtain a corpus which fits our needs. After preprocessing and selection there are 8,000 documents left.

Queries are the questions that are entered into search engine to retrieve results. In order to select queries, first we select n documents ($n \leq 30$) randomly. Then we select a portion of each document randomly and rewrite that portion as a query sentence.

3.2 Context Extraction

The context extraction part of our system, consists of two steps. The first step is analyzing the context with context analyzers. Context analyzers extract terms that represent context information from using three different context sources; Lucene [1], DBPedia-Spotlight [9], and Wordnet [12]. The second step is pairing. After the context extraction step, we pair terms and count pairs in order to find inter-term relevancy. Details of context extraction and pairing steps are explained in [6].

olcavtaner@isikun.edu.tr

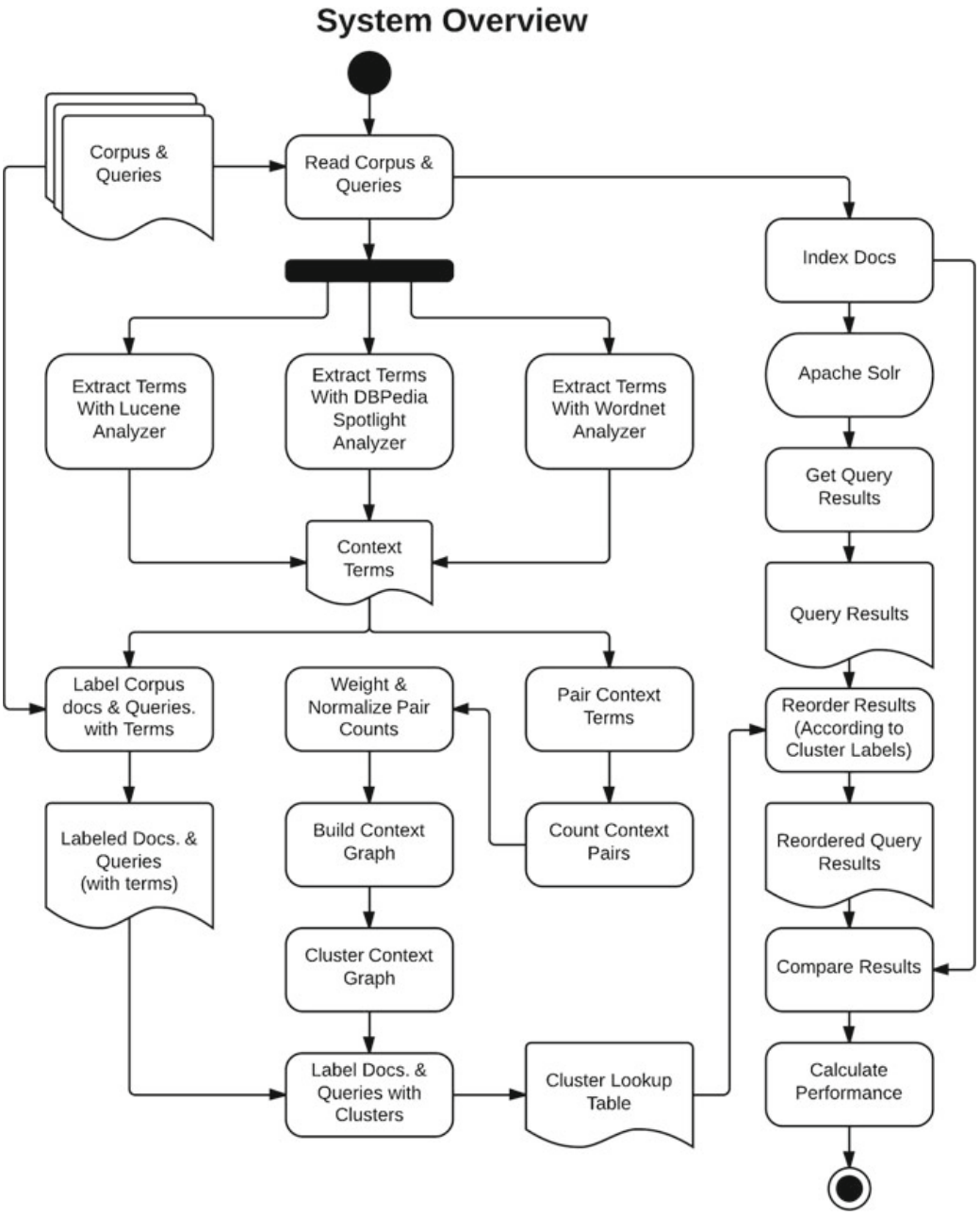

Fig. 1 System overview

3.3 Context Clustering

In order to cluster contexts, first, we build a graph of terms. Terms are connected via distances generated from pair counts that are previously calculated. Details of distance calculation explained in [6].

olcavtaner@isikun.edu.tr

Second step is clustering graph. In order to cluster our graph of terms, we use Markov Cluster Algorithm (MCL Algorithm) [3]. Finally, we export cluster context-term mapping which shows clusters that contain context terms.

3.4 Labeling and Scoring

In this part of our work, we label documents with clusters, in order to find which document belongs to which cluster. First, we index documents using SOLR [2] and retrieve results for the 30 queries which we have previously selected. Second, we label documents using context information that we extracted in context extraction step. Third, we also label queries using extracted context information. Next, we calculate new scores of documents. In order to calculate new scores we use dot product of cluster-counts of documents and queries. Then we sort results for each query according to new calculated scores.

4 Experiments

4.1 Setup

In order to measure performance, we need reference data to compare re-scored results. Therefore, we have designed and implemented a web application for data collection. The application retrieves top 50 results from SOLR without any extra scoring and shows these results to the user without ordering. The user can categorize the results into four categories. Then the application stores the user selections as scores. Each category has score between zero and three according to its degree of relatedness.

The categorized results of 30 different queries are obtained by our application and are given to several different users. Each query has a set of results for each user. Then we calculate means of scores that are given to each result of query. Then we reorder results according to new scores (Figs. 2, 3, and 4).

4.2 Results

In order to measure performance of our study, we use metrics that frequently used in the literature. These metrics are precision, recall, and f-measure. According to the experimental results, precision of Solr is slightly better. On the other hand, recall of our approach is a little better than Solr. In general, performance for both Solr and our approach are almost same. Precision, recall, and f-measure results also indicate similarities between performances.

olcavtaner@isikun.edu.tr

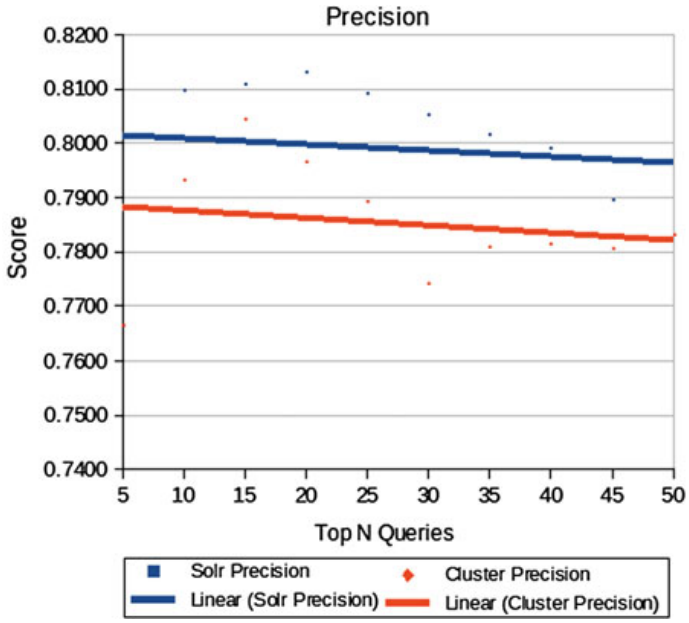

Fig. 2 Precision results of Solr and our approach

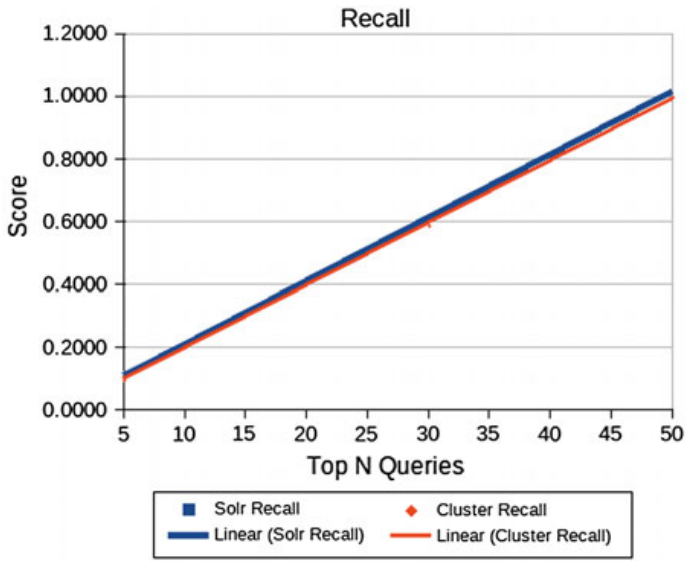

Fig. 3 Recall results of Solr and our approach

The corpus that we use, contains 8,000 documents. This size of corpus is small for a search engine. Using small corpus might be the reason that clusters have not formed good enough to distinguish concepts. Since concepts are not distinguished well enough, re-scoring with clusters do not change the results significantly.

olcavtaner@isikun.edu.tr

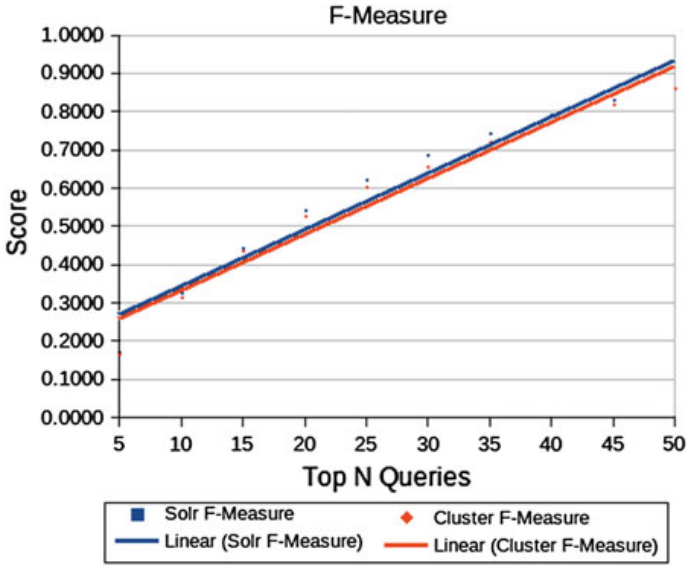

Fig. 4 F-measure results of Solr and our approach

5 Conclusion

In this paper, we focus on retrieving relevant data instead of processing big amounts of data. Llian's context over content approach [11] of human brain has inspired us to develop a method, which will improve search engine performance using a similar way to human thinking process. In our work, we propose a method which uses the context information of documents to improve the search engine performance.

According to the experimental results, we could not improve search engine performance significantly. The reason, mostly, is dependent to our corpus size. Because lack of computational power, we could use 8,000 documents which is very small set for a search engine. Using limited amount of documents is prevented forming clusters well enough. Re-scoring with using non well-formed clusters did not make any significant change.

Our content over context approach, that we presented, forms a basis for our future work. In the future, we plan to implement our design on a distributed system such as Hadoop so that we can process much bigger corpus. We also plan to extend context sources that we used to extract context information.

References

1. Apache lucene. http://lucene.apache.org/core/
2. Apache solr. https://lucene.apache.org/solr/
3. Mcl—a cluster algorithm for graphs. http://micans.org/mcl/
4. P.J. Brown, G.J.F. Jones, Context-aware retrieval: exploring a new environment for information retrieval and information filtering. Personal Ubiquitous Comput. **5**(4), 253–263 (2001)

olcavtaner@isikun.edu.tr

5. C.D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, Cambridge, 2009)
6. R. Duzagac, Improving search engine performance with context extraction using lucene, dbpedia-spotlight, and wordnet. Master's thesis, IK University (2014)
7. O. Egozi, S. Markovitch, E. Gabrilovich, Concept-based information retrieval using explicit semantic analysis. ACM Trans. Inf. Syst. **29**(2), 8:1–8:34 (2011)
8. A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic annotation, indexing, and retrieval. Web Semant. **2**(1), 49–79 (2004)
9. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, DBpedia—a large-scale, multilingual knowledge base extracted from wikipedia. Semant. Web J. **26**, 1–58 (2014)
10. L. Li, L. Zhong, G. Xu, M. Kitsuregawa, A feature-free search query classification approach using semantic distance. Expert Syst. Appl. **39**(12), 10,739–10,748 (2012)
11. R.R. Llinas, *I of the Vortex from Neurons to Self* (MIT Press, Cambridge, 2002)
12. G.A. Miller, Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
13. P. Mylonas, D. Vallet, P. Castells, M. FernÁndez, Y. Avrithis, Personalized information retrieval based on context and ontological knowledge. Knowl. Eng. Rev. **23**(1), 73–100 (2008)
14. M. Schuhmacher, S.P. Ponzetto, Exploiting dbpedia for web search results clustering, in *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, AKBC '13* (ACM, New York, 2013), pp. 91–96
15. C. Shaoul, C. Westbury, The westbury lab wikipedia corpus, edmonton, ab: University of alberta. (2010), http://www.psych.ualberta.ca/westburylab/downloads/westburylab.wikicorp.download.html

olcavtaner@isikun.edu.tr