2016 27th International Workshop on Database and Expert Systems Applications

Query Splitting For Context-Driven Federated Recommendations

Hermann Ziak
Know-Center GmbH
Inffeldgasse 13
8010 Graz, Austria
hziak@know-center.at

Roman Kern
Know-Center GmbH
Inffeldgasse 13
8010 Graz, Austria
rkern@know-center.at

***Abstract*—Context-driven query extraction for content-based recommender systems faces the challenge of dealing with queries of multiple topics. In contrast to manually entered queries, for automatically generated queries this is a more frequent problem. For instances if the information need is inferred indirectly via the user's current context. Especially for federated search systems were connected knowledge sources might react vastly differently on such queries, an algorithmic way how to deal with such queries is of high importance. One such method is to split mixed queries into their individual subtopics. To gain insight how a multi topic query can be split into its subtopics we conducted an evaluation where we compared a naive approach against a more complex approaches based on word embedding techniques: One created using Word2Vec and one created using GloVe. To evaluate these two approaches we used the Webis-QSeC-10 query set, consisting of about 5,000 multi term queries. Queries of this set were concatenated and passed through the algorithms with the goal to split those queries again. Hence the naive approach is splitting the queries into several groups, according to the amount of joined queries, assuming the topics are of equal query term count. In the case of the Word2Vec and GloVe based approaches we relied on the already pre-trained datasets. The Google News model and a model trained with a Wikipedia dump and the English Gigaword newswire text archive. The out of this datasets resulting query term vectors were grouped into subtopics using a k-Means clustering. We show that a clustering approach based on word vectors achieves better results in particular when the query is not in topical order. Furthermore we could demonstrate the importance of the underlying dataset.**

I. Introduction

Since the emergence of content-based recommender systems automated context-driven query extraction, which is closely related to just in time retrieval [1], is becoming increasingly popular. One of the challenges within this field is relevant context identification [2], since the performance of a content-based document recommender system is mainly determined by the quality of the initial extracted query, i.e. how well the query captures the potential user's information need. The presented work emerged from the EEXCESS (Enhancing Europe's eX-change in Cultural Educational and Scientific reSources) project. The project is open-source and can be obtained from GitHub[1]. The goal of this project is to recommend high quality content to users from a large range of different knowledge sources from the field of cultural heritage and scientific literature. In this setting it is expected that the query has not been explicitly stated by the user but is automatically derived from the user's context. Hence, the challenge in such a setting is to identify the topics, which might be of interest to the user. In settings like these literature suggests that it might be beneficial to cover multiple topics, since the actual interest of the user cannot always be correctly determined [1]. Additional strategies are to increase the diversity and novelty within results, increasing the chances that one of the presented results prove beneficial to the user. These techniques have shown to be of value to achieve user satisfaction, specially in a recommendation related context.

Therefore it might appear reasonable for a system that automatically infers a query from a user's context to merge multiple topics into a single query. While such a procedure will certainly work in many cases, there is a downside of including several topics within such automated queries, especially if the topics address independent aspects of the user's information need. Connected sources in federated retrieval systems might not respond well to such queries, especially if they are unusual long. This might lead to stations, where the connected source does not return any result at all. For instance, if the underlying search engine combines all query terms into a conjunctive query, dramatically lowering the chances of retrieving document containing all query terms. For that reason it might be beneficial if the query could be topically partitioned, specifically for such sources. In such a scenario the original query would first be split into topically coherent sub-queries, which would be individually submitted to the sources, resulting in multiple result lists. These result lists would then be aggregated in a single consolidated result list later on.

Most proposed approaches for query splitting or similar fields in literature appear to rely on two sources of information. Either they rely on initially retrieving documents or they make use of query logs. In a federated setting, consisting of multiple, independent connected sources, following a probing approach might introduce a high latency and thus cause longer response times. The second approach might also not appropriate for an federated search system. Particularly, in cases where the queries are automatically generated. Therefore it is expected, that the query logs will only contain the automatically inferred queries and will be heavily biased by the particular algorithm

[1] https://github.com/EEXCESS/recommender.git

2378-3915/16 $31.00 © 2016 IEEE
DOI 10.1109/DEXA.2016.37

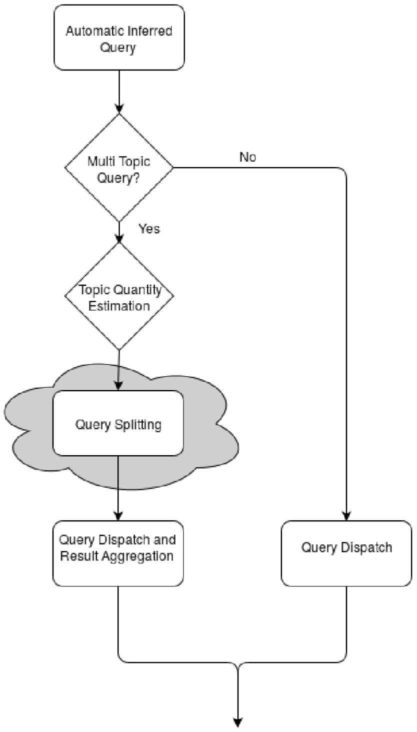

Fig. 1. Overview of the query processing pipeline. At first the query is tested whether it contains multiple, independent topics. If not, the query can be directly sent to the connected sources, otherwise the number of topics can be estimated. The focus of our work is then the splitting of a multi-topic query given the estimation of the number of topics.

on which the queries have been originally generated. Here an approach that avoids the additional latency and any unwanted bias would be preferred. Instead of making use of the information directly available to system (search results, query-logs), one can resort to the use of external knowledge sources. Examples for approaches that rely on knowledge bases are Word2Vec[2] and GloVe[3].

Both gained a lot of attention recently and proved to be helpful for a number of tasks, including the field of Information Retrieval. In this work we evaluate two approaches against each other and show that knowledge based approach, with the correct parameters like a dedicated trained vector-space model, has the potential to achieve adequate results without relying on the data from the actual source.

II. Related Work

Topical query splitting in the field of Natural Language Processing (NLP) is not yet extensively researched. In their work Yu et al. [3] proposed three approaches to split queries into different separated topics. Namely ‘Relevance-Feedback-Based Clustering‘, ‘Term-Based Clustering‘ and 'Document-Based Clustering'.
'Relevance-Feedback-Based Clustering' is an adoption of Boroding's [4] iterative method to group initially retrieved documents to query terms of the initial query. The top words of these groups are extracted from the documents content to create new sub queries with which the process is repeated.
'Term-Based Clustering' takes the ambiguity of query terms into account. Using Rocchio's query expansion algorithm to assign documents to each query term and calculate the cosine correlation to cluster the terms into a agglomerative hierarchical cluster. In the final step the cluster tree is cut into groups that represent the sub query.
'Document-Based Clustering' is similar to the Term-Based clustering but takes the diversity of the top retrieved documents into account.
All these three approaches rely on the preliminary retrieved top results according to the query. Other approaches, that are not directly targeted on query splitting but on query topic detection, rely on the use of query logs and try to identify latent topic [5], [6]. In the related field of query segmentation the usage of external knowledge bases has already proven to be effective [7]. Query segmentation typically does not split a query but tries to identify important phrases and concepts within the query.

Parts of this work rely on Word2Vec, which received a lot of attention recently, as it provided good results in many text mining scenarios. Word2Vec is a tool that can be used to provide an bag-of-words based vector representation of words and phrases. Here an text corpus is taken as input to extract a vocabulary of which vectors of the relation of this vocabulary are learned by a predictive model [8], [9]. In contrary for the second word vector based approach we rely on "Global Vectors for Word Representation" (GloVe) [10] which is similar to Word2Vec but based on a count-based model instead of a predict model. A comprehensive comparison of this two concepts can be found in the work of Baroni et al. [11].

The in this work presented evaluation does not directly addresses the estimation of the amount of topics within the query and relies on prior knowledge which is needed for both approaches; the amount of covered topics. Nevertheless one can refer to the estimation approach of Tibshirani et al. [12] by the use of gap statistics and the more recent work of Pham et al. [13].

III. System

The motivation for the work presented in this paper is the scenario of a federated recommender system in an uncooperative setting. Here the uncooperative property of the connected sources stems from the fact, that these source cannot be changes or their behaviour cannot be altered, apart from the submitted query. Thus, the query needs to be designed in a way that it optimally matches the individual sources, which might be vastly different to each other. For example, some sources combine query terms via an OR operator (disjunction query), while others combine query terms via an AND operator (conjunction query). Some sources even sport a more sophisticated query analysis and processing process.

[2] https://code.google.com/archive/p/word2vec/
[3] http://nlp.stanford.edu/projects/glove/

The target of the EEXCESS system is to support users from the cultural heritage domain, thus the majority of the sources connected to the federated recommender contain content like museum objects, e.g. pictures of coins. Apart from the federated recommender, the EEXCESS system consists of a number of other components to achieve its goal. At first, there are a number of components to interact with the users, for example a browser extension. This components continuously tracks the user interactions with the browser, this the goal to i) precisely assess the current user's information need, and ii) provide means to display recommendation to the user, without obstruction the user's work flow. Given the user's consent, the history of the user is tracked, i.e. the sequences of visited Web pages. To improve the automatic query generation process, the currently opened Web page is analyzed in more detail. This task is conducted by the C4[4] library. Here an algorithm tries to detect the paragraph, which is most likely to be currently read by the user. Next another library, the so called DoSeR[5] service [14], is invoked to analyze the current paragraph, which is then used to fully automatically formulate a query.

This query should reflect the content of the user's current context. In many cases this context may be composed out of a number of different topics, that might either be closely related or only loosely related to each other. The query is then submitted to the federated recommender component[6]. The task of this component is to process the query further and to distribute the query to all connected sources, collect the results and to combine all results into a single aggregated result list. In order to achieve its goal for each connected source, there are additional adapter components, called partner recommender. These adapter components are responsible to transform and translate the query in a way that the sources will be able to generate optimal results. In order to achieve its goal, the adapter may need to split multi-topic queries into separate queries, which are processed independent from each other.

IV. Query Splitting

The main goal of the presented work is to get an insight into the impact of different query splitting algorithms with a focus of their use within a federated search setting. An schematic overview of the query processing pipeline is given in Figure 1. The priority here is to create solution that is neither depending on the query extraction algorithm, which might change over time, nor makes use of probing of the partners which adds additional processing time. In order to achieve this, we compare two vastly different approaches with each other: a) a very simple approach, and b) a more sophisticated classification based approach, making use of algorithms, which are currently considered as state-of-the-art in many text mining tasks. Within both approaches the only preprocessing step that is applied on the initial queries was stop word removal.

a) Split Approach: The first approach is the most obvious solution. Here the query is just split into groups of equal length of query terms according to the amount of joined queries. This approach seems only to be feasible in a setting where topics within the joined queries are evenly distributed. The results obtained with such an algorithm should give a basic understanding of how hard the task of query splitting is. In order for such a simple approach to produce meaningful results we designed part of the evaluation in such a way that this algorithm produce close to optimal results.

b) Word Vector Classification Based Approach: The second approach is based on the Word2Vec and on the GloVe algorithm. For the Word2Vec based evaluation we relied on the already pre-trained Google news model. The model contains a 300-dimensional vectors for about 3 million words extracted from the Google News dataset[7]. It can be expected that the approach would benefit from a dedicated trained model closer to the domain of the queries. The dataset[8] utilized for the GloVe based evaluation is a fusion of two different datasets. On one hand a Wikipedia dump [9] of 2014 and on the other hand the English Gigaword Fifth Edition [10] containing newswire text data. For each query term within the query the word vectors were extracted from one of the models. To also take the position into account these vectors were expanded by the position of the word within the query. The vectors were then used to group the according query terms with the well known k-means algorithm [15]. K-means was initialized with the number of cluster centroids according to the chosen amount of queries merged.

V. Evaluation Setup

Since dataset containing content-based automated context-driven queries are difficult to obtain we decided to rely on an already well studied dataset. The Webis-QSeC-10 training set [16] containing about 5000 queries extracted out of query logs. The queries within the query set are user formulated queries that were further annotated highlighting segments within each query via crowdsourcing. Although the dataset contains user formulated queries it should be still adequate to gain insights about the performance of the two proposed approaches. Another motivation for using this data-set is the fact, that it contains only a limited amount of named entities as multi-term expressions. Often queries will contain such named entities, which can be addressed by dedicated mechanism to detect such expressions. For this particular data-set such tools are not strictly needed, allowing us to study the isolated impact of the query splitting algorithms.

In the first evaluation setup N unrelated queries of the dataset are joined, were as N ranges from 2 to 4. In this first setup we assume that the queries send to the system are in a topical sequence. For example in a context-driven query

[4] https://github.com/EEXCESS/c4

[5] https://github.com/EEXCESS/DoSeR

[6] https://github.com/EEXCESS/recommender

[7] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

[8] http://nlp.stanford.edu/data/glove.6B.zip

[9] http://dumps.wikimedia.org/enwiki/20140102/

[10] https://catalog.ldc.upenn.edu/LDC2011T07

TABLE I

Rand Index results of the two splitting approaches within the first scenario where the position is taken into account. The values descent from iteration to iteration for each of the approaches.

	Two Queries	Three Queries	Four Queries
Word2Vec Kmeans Rand Index	0.710	0.643	0.595
GloVe Kmeans Rand Index	**0.729**	**0.697**	**0.648**
Split Approach Rand Index	0.717	0.664	0.631

TABLE II

V-Measure results of the two splitting approaches within the first scenario where the position is taken into account. The V-Measure values seem to be almost independent from the amount of mixed queries.

	Two Queries	Three Queries	Four Queries
Word2Vec Kmeans V-Measure	0.770	0.770	0.769
GloVe Kmeans V-Measure	**0.788**	**0.806**	0.780
Split Approach V-Measure	0.775	0.781	**0.789**

extraction approach based on paragraphs it might happen that two paragraphs are falsely unified and detected as only one. (e.g Two different article snippets on a news page detected as being just one snippet.) In that case the extracted query terms would consist of topically unrelated groups but might still be in the correct sequence. Therefore the queries are kept in their original order in the approach as well.

The second evaluation setup differs from the first setup by the assumption that the query terms might no be send in a topically related order. This might happen for example in a system that extracts keywords out of a paragraph and returns a weighted list of important query terms. In this case the query terms might have several mixed topics which can not be distinguished only by their order. Therefore the already joined queries were randomized. Again both approaches were applied on the resulting list of query terms.

Both evaluation setup were tested with all queries in Webis-QSeC-10 combined in group of two queries, three queries and four queries.

We chose two measures to evaluate the performance of the algorithms: Rand index [17] and the more recent V-measure [18]; both are common measures used to evaluate clustering performance. Where the Rand Index returns the similarity of two clusters by considering all pairs of samples and the V-measure represents the harmonic mean between homogeneity and completeness of two clusters. Rand Index returns values from -1 to 1 where 0 could be considered as totally random labeling were as V-measure values from 0 to 1

VI. Results

Table I and Table II shows the results of the setting where the queries are in their topically related order. V-measure seems to be constant no matter how many queries are joined although there seem to be minimal improvement on the query splitting approach. The results of the Rand Index measure seems to get smaller by each sub query added to the joined query for both approaches.

TABLE III

Results of the two splitting approaches within the second scenario where the position is not taken into account. The Rand index seems to decline with each iteration. The highest values are achieved by the GloVe based approach. The Rand Index results of the splitting approach indicate a random assignment to the clusters as expected.

	Two Queries	Three Queries	Four Queries
Word2Vec Kmeans Rand Index	0.088	0.071	0.056
GloVe Kmeans Rand Index	**0.281**	**0.232**	**0.199**
Split Approach Rand Index	0.008	0.003	0.000

TABLE IV

Results of the two splitting approaches within the second scenario where the position is not taken into account. Here highest values are produced by the Glove based approach, which even raise by each iteration. For the Word2Vec based approach the values seem to be almost stable independent from the amount of mixed queries. The lowest results are achieved by the splitting approach.

	Two Queries	Three Queries	Four Queries
Word2Vec Kmeans V-Measure	0.373	0.341	0.373
GloVe Kmeans V-Measure	**0.427**	**0.477**	**0.502**
Split Approach V-Measure	0.278	0.267	0.236

Table III and Table IV presents the results of the randomized joined queries. Here the results for the splitting approach are low in general. In particular the results Rand Index indicate totally random behavior. The Word2Vec approach seems to work better in this setting on a quantifiable level. Here V-Measure from the Word2Vec dataset seems to be almost independent from the amount of queries mixed whereas the values for the Splitting approach decline. The highest figures in general are achieved by the Glove dataset based approach where the V-Measure values rises with the number of topics.

VII. Discussion

Given the results it seems that in the first scenario there is one dominant factor. The position of the split within the query seems to contain the most information. This can be explained by the distribution of the query length within the dataset. Arampatzis et al. [19] showed in their study that the majority of user queries contain usually between two to five keywords. This seems to be the case within the Webis-QSeC-10 training set as well [7]. Therefore the average split might often occur on the correct position or just occur off by one. In such a setting, the information introduced by the Word2vec approach seems to be not particularly beneficial and, to an extend, even seems to have an impact on the performance. In comparison the GloVe model achieved the best results in almost all cases. This might either be due to the fact that the used Word2Vec model missed several named entities within it's dataset or due to the underlying concepts of Word2Vec and GloVe.

The second scenario demonstrates the impact of an assumption that the query might be in a topical sequence. Here it is apparent that it is not sensible to just simply split queries after a certain amount of query terms when the query is not in a topical sequence. The word vector classification based approach, seems to perform better here and is able to split the query terms correctly in a number of cases. Still improvements to the algorithm should be easy to obtain, for example by using a dedicated Word2Vec or GloVe model.

VIII. Conclusion And Future Work

In general a naive splitting approach seems only to be reasonable in cases where the amount of topical groups in query terms can be reliably estimated and their correct order can be inferred. When this is not the case the split will most properly lead to unrelated items and poor results in general.

Word2Vec and GloVe are producing comparable results if these assumption are met. This approach seems to be feasible also in cases of query terms not being in topical order. Still this approach also allows room for improvement.

As a logical consequence, in future work we plan to train dedicated Word2Vec and GloVe models both based on the same sources (e.g data of a query logs, knowledge bases and news wire text). In that regard including data like the titles of Wikipedia pages could be beneficial to cover named entities as well. Using the Wikipedia titles has been demonstrated as a valuable approach in the closely related field of query segmentation [7].

We plan to evaluate which optimizations regarding feature sets could improve the results by the use of different clustering methods. Furthermore, it might be beneficial to gather a dedicated dataset closely resembling the behavior of automatic query generation algorithms.

Acknowledgment

The presented work was developed within the EEXCESS project funded by the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 600601. The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[1] B. J. Rhodes, "Just-in-time information retrieval," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[2] J. Schlötterer, C. Seifert, W. Lutz, and M. Granitzer, "From context-aware to context-based: Mobile just-in-time retrieval of cultural heritage objects," in *Advances in Information Retrieval*. Springer, 2015, pp. 805–808.

[3] X. Yu, F. A. D. Neves, and E. A. Fox, "Hard queries can be addressed with query splitting plus stepping stones and pathways." *IEEE Data Eng. Bull.*, vol. 28, no. 4, pp. 29–38, 2005.

[4] A. Borodin, L. Kerr, and F. Lewis, "Query splitting in relevance feedback systems," 1968.

[5] Q. Ye, F. Wang, and B. Li, "Starrysky: A practical system to track millions of high-precision query intents," in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 961–966.

[6] X. He, J. Yan, J. Ma, N. Liu, and Z. Chen, "Query topic detection for reformulation," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 1187–1188.

[7] M. Hagen, M. Potthast, A. Beyer, and B. Stein, "Towards optimum query segmentation: in doubt without," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 1015–1024.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[10] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[11] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors." in *ACL (1)*, 2014, pp. 238–247.

[12] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[13] D. T. Pham, S. S. Dimov, and C. Nguyen, "Selection of k in k-means clustering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 219, no. 1, pp. 103–119, 2005.

[14] S. Zwicklbauer, C. Seifert, and M. Granitzer, "Doser – a knowledge-base-agnostic framework for disambiguating entities using semantic embeddings," in *Proc. European Semantic Web Conference (ESWC)*, 2016.

[15] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[16] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam, "Query Segmentation Revisited," in *20th International Conference on World Wide Web (WWW 11)*, S. Srinivasan, K. Ramamritham, A. Kumar, M. Ravindra, E. Bertino, and R. Kumar, Eds. ACM, Mar. 2011, pp. 97–106.

[17] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[18] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 410–420.

[19] A. Arampatzis and J. Kamps, "A study of query length," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 811–812.