Query Splitting For Context Driven Federated Recommendations

How to deal with long, automatically generated queries

Hermann Ziak, **Roman Kern**

Know-Center - Research Center for Data-Driven Business and Big Data Analytics

TIR 2016

Introduction

Context

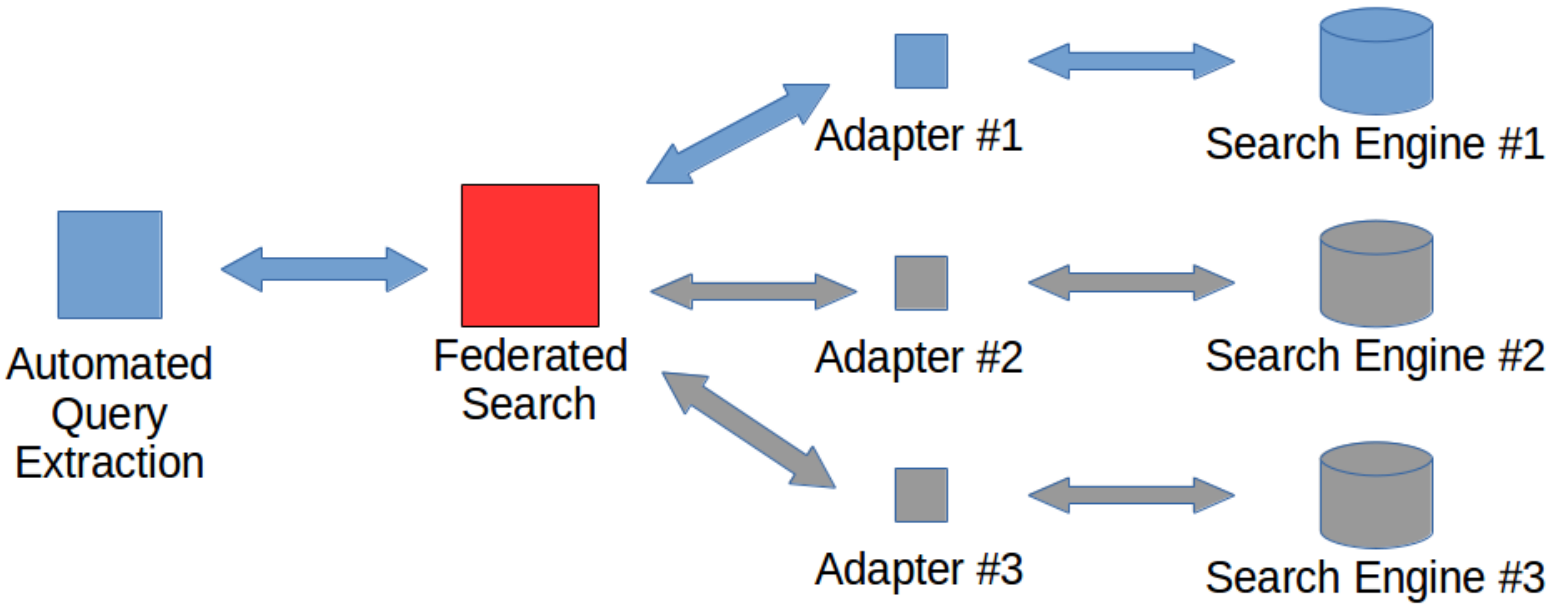
The Ecosystem

Cross-vertical federated aggregated search in an uncooperative setting for queries automatically extracted from an user's context.

The Goal

Provide a solution for search engines that do not deal well with long, multi-topic queries.

Context


Automated
Query
Extraction
Federated
Search
Adapter #1
Adapter #2
Adapter #3
Search Engine #1
Search Engine #2
Search Engine #3

Context

Federated Search

Meta search where for each query is forwarded a number of search engines (also called sources), typically remotely located.

Uncooperative Setting

The search engines do not provide information and cannot be tweaked, basically assumed to be black boxes.

Context

Cross-Vertical Search

Combining results from different types of search engines, for example combine textual search results with image search results.

Aggregated Search

Combine multiple search results into a single, consistent search result list.

Motivation

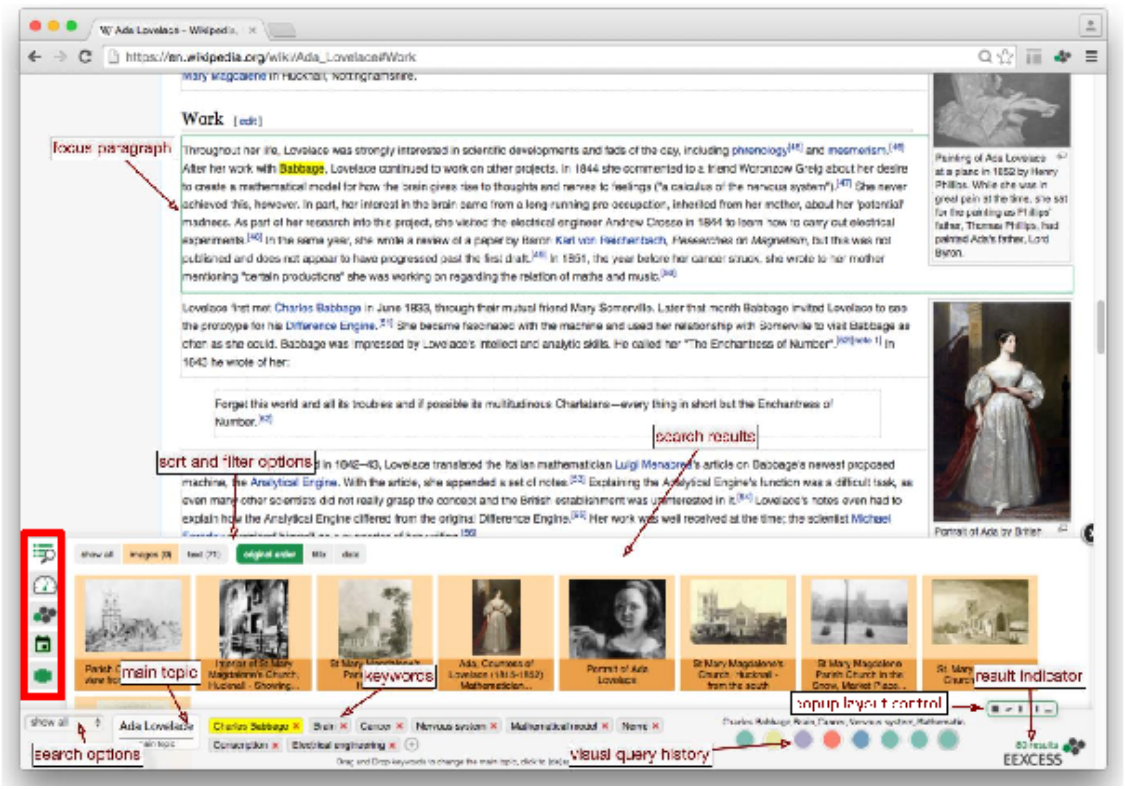
Context-Driven Query Extraction

- Queries are automatically generated
  - … so that the user does not have to type in a query
  - … instead, it is inferred from the user current context
- Just-In-Time Information Retrieval
  - Close connection to Recommender Systems

Motivation

Example

- A user is browsing the **Wikipedia**
  - Reading a paragraph of one of the articles
  - **Key terms** from that very paragraph are extracted
  - Sent to a **search engine** to gather results



Motivation

Consequences

- A query might contain multiple parts
  - … might cover different, unrelated aspects, i.e. multi-topic
  - thus might reflect independent information needs
- Queries might be long
  - … to alleviate to miss important terms
  - Since the actual information need of the user cannot be always be correctly determined

Note: Additionally in such scenarios, often the search result list also *artificially* includes diversified and novelty.

Motivation

Problem

- Some search engines do not fare well with multi-topic queries
  - E.g. is the search engine only supports conjunction queries
    - ⋆ … will yield empty search results most of the time
- Some search engines do not fare well with long queries

Approach

Proposed Approach

Topical query splitting:

1. Take a long query
2. Identify the coherent parts
3. Reformulate individual queries for each of these parts
4. Issue searches for each of the sub-queries
5. Combine the results into a consistent search result list

State-of-the-Art

Main approaches to query splitting

1. Relevance feedback based
   - Multiple requests to the search engines
   - Initial search results are analysed
2. Use of query-logs
   - Identify common sub-queries

... both approaches are not well suited for our setting (no access to query logs, high latency, ...)

Related Problems

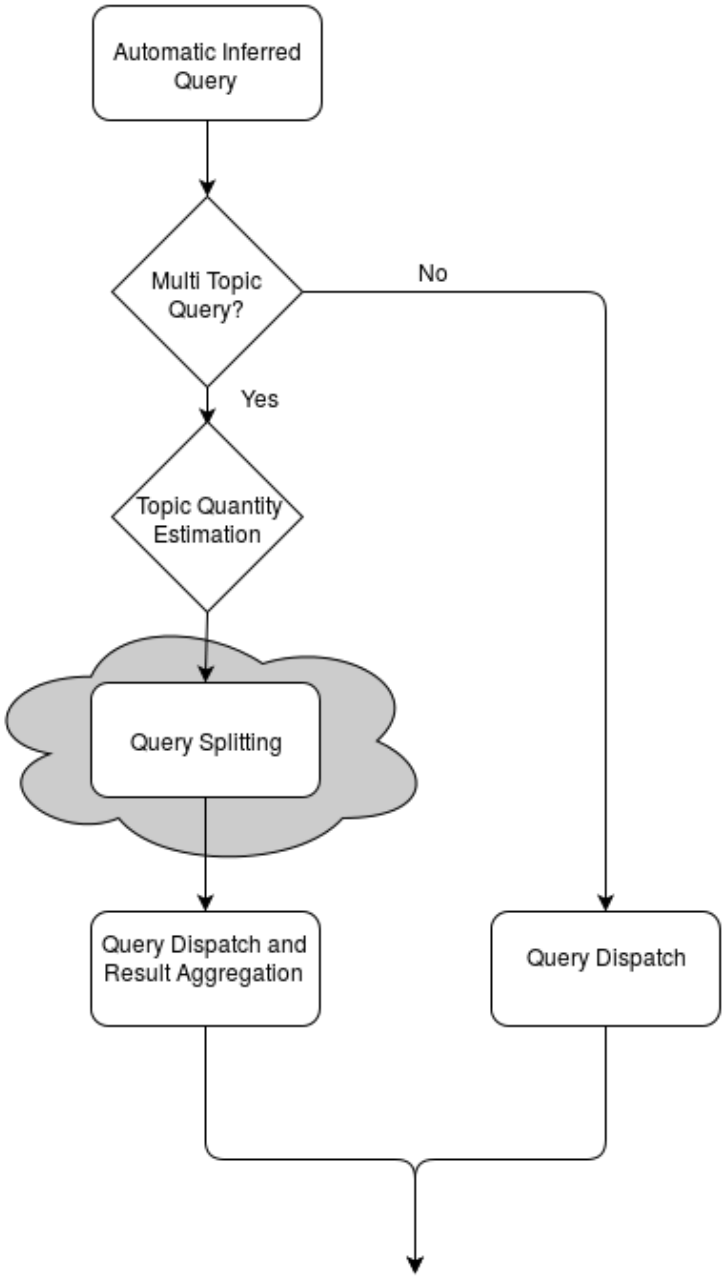
Query Topic Detection

- Identify the (latent) topic(s) of the query
  - Often uses query-logs

Query Segmentation

- Identify phrases and (multi term) named entities within a multi-term query
  - Recent approaches make use of external knowledge-bases
  - For example:
    - [new york city griffith building]
    - [“new york city” “griffith building”]

System Overview



Method

Constraints on the query splitting

- Does not use relevance feedback or query-logs
- Works independent from the query extraction method
  - … as it may change/evolve over time

Proposed Method

Our query splitting method

1. Introduce a similarity function
   - Allows to compute the pairwise similarity between query terms
   - Makes use of a word embedding function
     - ★ Transforms a word into a vector representation
     - ★ Makes use of an external knowledge base
2. Apply a clustering algorithm
   - Each cluster represents a single topic, i.e. sub-query

Word Embedding Methods

Word2Vec

- Gained a lot of attention recently
  - Makes use of neural networks
- Trained using the context of terms
  - Requires training corpus
- Similar terms end up close to each other
  - Even allows vector arithmetic in the projected space
  - … can be used to answer questions like:
  - “Athens is to Greece as Rome is to […]”
- We used the default parameters and the already provided pre-trained model
  - Google news, 300 dimensions, 3 million words

Word Embedding Methods

GloVe

- Conceptually similar to Word2Vec
- Optimises distances within the projected space
  - Based on co-occurrence statistics
- Requires training corpus
- We used the default parameters and an pre-trained model
  - Merge of a Wikipedia dump (2014) and the English Gigaword corpus (5th edition)

Clustering Methods

k-Means

- Simple, but well performing clustering method
- Requires the number of clusters being specified in the beginning
- Requires a distance/similarity function

Evaluation

Input Dataset

Base Dataset

- Webis-QSeC-10 training set
  - Originally developed for query segmentation, not query splitting
  - Need a strategy to evaluate query splitting
- Consists of 5000 queries extracted from query-logs
  - … further annotated
- The dataset contains only a limited amount of named entities
  - … well suited for our task

Evaluation Dataset

Run #1

1. Select a number of topics
   - Out of [2, 3, 4]
2. Pick corresponding many queries from the base dataset
3. Concatenate the queries into a single query

Run #2

1. [equal to run #1]
2. [equal to run #1]
3. Randomly construct a query from the query terms

Tasks

- Reconstruct the original queries

The Baseline

Baseline Algorithm

- Split the query into equally long segments
  - … according to the correct number of topics
- Very simple baseline

Evaluation Measures

- Rand Index
- V-Measure

Results

Results for run #1 (V-Measure)

Algorithm	2	3	4
Baseline	0.77	0.78	**0.79**
Word2Vec	0.77	0.77	0.77
GloVe	**0.79**	**0.81**	0.78

Results

Results for run #2 (V-Measure)

Algorithm	2	3	4
Baseline	0.28	0.27	0.24
Word2Vec	0.37	0.34	0.37
GloVe	**0.43**	**0.48**	**0.50**

Discussion

- *Surprisingly*, GloVe consistently outperforms Word2Vec
- Baseline performs very well in run #1
  - Due to the sub-queries being of similar length
  - i.e. most errors are just off-by-one errors
- Performance should improve with training data sets close to the domain
- Depends on the actual setting, if run #1 or run #2 resembles the true behaviour

Conclusion & Future Work

- Some form of query splitting necessary for certain search engines
  - But, should be combined with:
    - ★ Query segmentation - to detect named entities and phrases
    - ★ Algorithm to detect the number of topics, especially single topic queries
  - Additional information from query extraction could be beneficial as well
- Open issues
  - Evaluation of the aggregates search result
  - In vivo evaluation of the whole system

The End

Thank you for your attention!

Acknowledgements

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.