D5: Databases and Information Systems
Advanced Topics in IR, SS 2016
Dr. Vinay Setty and Dr. Jannik Strötgen
Assignment 5b, due: **30 Jun 2016**

max planck institut informatik

Please submit your solution as a PDF and source code together in a single tar/zip file, name your pdf file in the following format: "Lastname-Immatrikulationnumber-AssigmentX.pdf", also include Immatrikulationnumber in the title of your document and email it to `atir16@mpi-inf.mpg.de` *before the due date mentioned above!*

Temporal Query Classification (15 Points)

Problem 1.
Assume that you have access to a large query log with entries of the form

u167823 fifa world cup 1990 2015/01/11:10:46:22

telling you which user issued which query at which time. With this, how would you

a) identify implicitly temporal queries

b) distinguish between atemporal, temporally unambiguous, and temporally ambiguous queries?

Non-Redundant Indexing (15 Points)

Problem 2.
The non-redundant indexing method by Zhang and Suel (2007) breaks up documents into smaller segments and indexes them using a two-level index structure. The first level maps words to segments, so that one can find segments that contain a specific word; the second level maps segments to documents, so that one can find out which documents contain the previously identified segments. While ideal for web archives, the method does not support time-travel text search, i.e., evaluating a query "as of" a given time in the past.
How would you modify the two-level index structure, so that it supports it? Please also discuss how you would process time-travel text queries on the modified index structure.

Cross-Time Information Retrieval (20 Points)

Problem 3. Search in historical document collections needs to bridge a vocabulary gap between queries issued by today's user and old documents written in archaic language. In the lecture we saw one example (Koolen et al., 2006) how language evolution (e.g., different spellings) can be dealt with. Terminology evolution is a highly related problem: sometimes entities change their name (e.g., Saint Petersburg was known as Leningrad thirty years ago) or new concepts drive away old ones with similar functionality (e.g., walkman/ipod, fax/e-mail, altavista/google).
Assuming that you have a large collection of timestamped documents at hand (e.g., a newspaper archive), devise a similarity measure sim which allows you to compare two words u and v, in their semantics, when used at times (e.g., during a specific year) t_u and t_v, respectively. For instance, your similarity measure should indicate high similarity sim(walkman, 1987, ipod, 2007) between walkman when used in 1987 and ipod when used in 2007.

Programming Assignment (25 Points + 25 Points from Assignment 5A)

Problem 4. Use the temporal expressions you have extracted from the document collection (or a subset of the document collection) in the context of assignment 5A to extend your search engine in such a way that temporal information needs can be addressed. Elasticsearch supports a type date so that it is probably a good idea to add the normalized temporal expressions (i.e., the values of TIMEX3 tags) as a separate field to your existing index. Please just index all the temporal expressions of type "date" and ignore all extracted time, duration, and set expressions. Of course, you have to use the normalized "value" information.
If you search for how elasticsearch can handle temporal queries, you will find information on the type "date" for indexing, and on "range queries". Please try to extend your search engine in such a case that temporal queries (if possible range queries) can be processed. Please describe (i) how you ran HeidelTime on the documents (assignment 5A), (ii) how you index the normalized temporal expressions, (iii) how you realize the query functionality, and (iv) if you handle temporal expressions of different granularity all in the same way or not. Please also describe whether you experienced any problems during this process. In addition, please describe how the ranking is realized or could be realized.