Weekly reading report

This week I read two survey papers and a few papers I feel interested. [1] from Netherlands group does a good survey on query auto completion papers that has been published before 2016, which classify existing QAC approaches into two broad categories: heuristic models and learning based models. And these two categories can be split into three groups: time-sensitive, contextual-based and demographic-based (previous search history and user profile information). Except query popularity, time series is also used by systems (see next section). [2] is a short survey on some papers published in 2017 about efficiency in IR. It mentions a static cache that stores both individual posting lists and bigram intersections of posting lists. Also a paper in score-at-a-time search engine stated an interested fact that modern CPU suggests no performance improvement on some compressed posting lists. [3] gives an insight on how to distribute shards across machines to address throughput and latency trade-offs. [4] examine indexing techniques for repetitive collections. [5] introduce Waves, a search engine to do multi-tier retrieval (examine documents in a given tier before moving on to the next).

1 QAC survey paper

Since I am focusing on QAC, this section recaps the key points in [1]. Based on query popularity, the straightforward method is MLE based on frequency or most-popular-completion (MPC) as referred by Bar-Yossef. In real-time news search, time plays much more important role so that long term query log are not sensitive to these applications. Using time-series analysis, we are able to put emphasis on "trending" keywords in QAC. Methods based on time-series use temporal trends, periodicity parameters with smoothing applied. Research in this direction tries to detect "bursts" in query frequencies of recent query log. In addition to time-based clues, similarity between candidate and query is also taken account, those include semantic similarity, temporal patterns.

Search context can also be used to personalize QAC, this involves aggregating list of query logs in order to compare their hybrid similarity to search context. Furthermore, user's implicit negative feedback such as skipped query completion, eye contact can also be used to do QAC feedback. This class of methods is called "user-

centered" QAC in that survey paper. Finally, learning-based approaches such as Likelihood ratio (LLR), learning from user action as features (e.g. convolutional latent semantic model), online learning, query entity discover [7], diversification, click models, etc.

As for implementation, QAC requires efficiency since user input requires responsive actions. Data structure such as Trie, prefix/postfix tree. In the end, This paper also compare the subtle differences between query suggestion, query expansion and query correction.

2 Collaborative Filtering (CF)

CF used to build recommendation system by computing rating that each user has input. [6] addresses the problem of current evaluation method not investigating the temporal characteristics of produced recommendations, researcher has no idea if the system is recommending the same item to users over and over again, thus results tend to stagnate and not interesting to users.

3 Temporal Query

[8] explores how queries and their associated documents change over the course of 10 weeks by analyzing query log data. The change in which pages were relevant to query during different time periods (such as sports in its season) reflects people's query intent was also changing. This paper uses features such as changes in query popularity to model changes in latent entity or user's query intent. [9] proposes latent entity space (LES) to represents entity by vector space and sue a probabilistic framework to model the relevance in this entity space.

4 Reference

[1] A Survey of Query Auto Completion in Information Retrieval
[2] Efficiency in information retrieval: introduction to special issue.
[3] Efficient distributed selective search.
[4] Document retrieval on repetitive string collections.

[5] Waves: A fast multi-tier top-k query processing algorithm.
[6] Temporal Diversity in Recommender Systems.
[7] Named Entity Recognition in Query.
[8] Understanding Temporal Query Dynamics.
[9] Latent entity space: a novel retrieval approach for entity-bearing queries.