Senior Thesis Prospectus

An Experimental Study of Multi-stage Retrieval Systems

Mohammed Yusuf Ansari

Advisor: Mohammad Hammoud

September 2019

Abstract

Information Retrieval (IR) is concerned with search over large unstructured data like web pages, emails, and image libraries, among others. An IR system is able to search over a large collection of data because it pre-processes and stores such data in a space- and time-efficient inverted index data structure. In an IR system, the process of retrieving relevant data (referred, henceforth, to as documents) for a given user query is done in three major stages, namely: (1) *candidate set generation stage*, (2) *feature extraction stage*, and (3) *candidate set re-ranking stage.*

During the candidate set generation stage, a retrieval strategy with a pruning model (e.g., WAND [5]) is executed on top of an inverted index to retrieve top K ranked documents for the given user query. In the feature extraction stage, various features are extracted from the top K documents generated by the first stage. In the candidate set re-ranking stage, the features extracted by the feature extraction stage are passed to a trained machine learning model, which re-ranks the top K documents. After candidate set re-ranking is complete, top 10-50 results are returned as a final answer for the given user query.

In this thesis, we will comprehensively investigate and analyze the correlation between the different stages of end-end multi-stage IR Systems. In particular, we will pose two research questions: (1) what are the effects of stages one and two on stage three? and (2) are there features in stage two that can be leveraged in stage one to boost the end-to-end efficiency and effectiveness of multi-stage IR systems?

We suggest that the above two research questions can be answered through a comprehensive experimental study, which observes, identifies, explains, and potentially models the performance of a representative search engine under various configurations. Based on the gathered results and insights, we will provide recommendations for best practices and promising research directions.

Problem Significance

It is critical for an Information Retrieval (IR) system to be *efficient* (i.e., quick) and *effective* (i.e., accurate). To exemplify, search engines like Google, Bing and Baidu are deemed as a popular application of an *efficient* and *effective* multi-stage IR system. If a search engine takes too long to respond or returns irrelevant results to a given user query, it will be considered inefficient or ineffective, and consequently fail in fulfilling users' information needs.

To improve the *efficiency* of an IR system, new retrieval strategies, which involve sophisticated pruning models have been developed over the past few years. Researchers have also created machine learning models to predict the optimal size of a candidate set using pre-retrieval static features [2]. These retrieval strategies and machine learning models served tremendously in increasing the efficiency of the candidate set generation stage, while maintaining its effectiveness.

To maximize the system's effectiveness, new machine learning techniques (e.g, Gradient boosted regression trees [8], lambda-mart [7], etc.,) have been developed for the candidate set re-ranking stage. Alongside, more features were added to the feature extraction stage in an attempt to improve the effectiveness of these machine learning algorithms [9].

We note, however, that these machine learning algorithms were developed in complete obliviousness to the configuration of the candidate set generation stage (e.g., retrieval strategy, size of candidate set, etc.,). For instance, it is reasonable to assume that the quality and number of documents retrieved in the first stage would impact the performance of the machine learning models in the candidate set re-ranking stage. Moreover, the type and number of features extracted in the second stage may influence the effectiveness of the third stage because different machine learning models may perform differently depending on the input features. To this end, it is important to realize that optimizing configuration parameters (e.g., size of candidate set, number of features, etc.,) in stage 1 or stage 2 without considering stage 3 may not result in optimizing the end-end effectiveness and efficiency of a multi-stage IR system.

We believe that understanding the performance correlation between the above three stages would serve considerably in improving the end-end effectiveness and efficiency of IR systems. To our knowledge, this correlation and its real impact on the user experience has not been fully investigated in literature. We aim to fill this critical gap via empirically identifying the optimal configuration parameters of stages 1 and 2 for each state-of-the-art machine learning model in stage 3.

Related Work

There has been a significant effort in the past few years to increase the efficiency of the candidate set generation stage and the effectiveness of the candidate set re-ranking stage. Researchers have optimized parameters in each stage and measured the resultant impact on the effectiveness and efficiency of the underlying end-end retrieval system. We next summarize some of the work that has been conducted in the field.

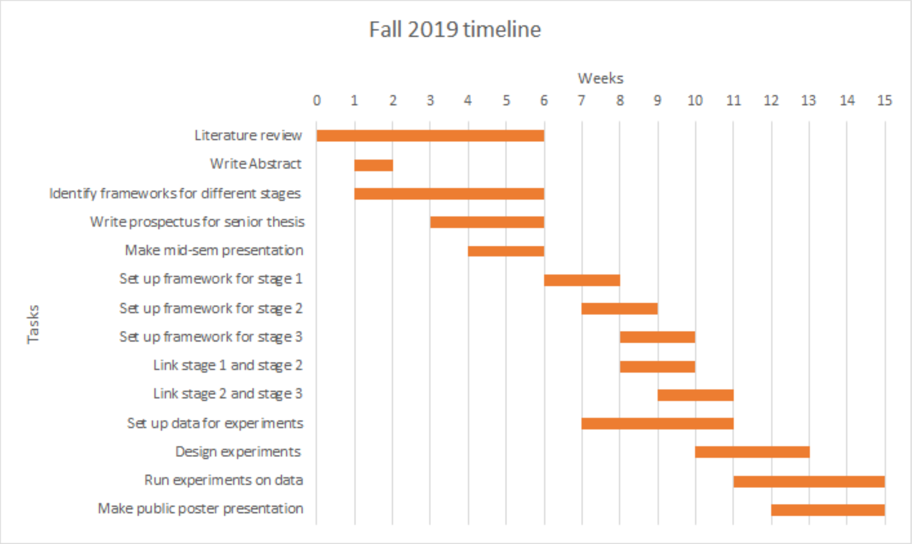
Analyzing the effect of different retrieval strategies on end-end effectiveness:
N. Asadi and J. Lin [1] analyzed effectiveness/efficiency tradeoffs with three candidate generation approaches: postings intersection with SvS [1], conjunctive query evaluation with WAND [5], and disjunctive query evaluation with WAND as well. Fixed configurations of the feature extraction and candidate set re-ranking stages were used to isolate the end-to-end effectiveness and efficiency implications of different algorithms. Based on their experiments, N. Asadi and J. Lin concluded that postings intersection with SvS is the most efficient candidate set generation approach. However, SvS was shown to have low end-end effectiveness as compared to conjunctive and disjunctive WAND. This result highlights the importance of term and document frequencies for the candidate set generation stage. Independently, the paper also concluded that conjunctive WAND is a better candidate set generation algorithm because it is much faster than disjunctive WAND in query evaluation. Moreover, in terms of end-to-end effectiveness, it is statistically distinguishable from disjunctive WAND.

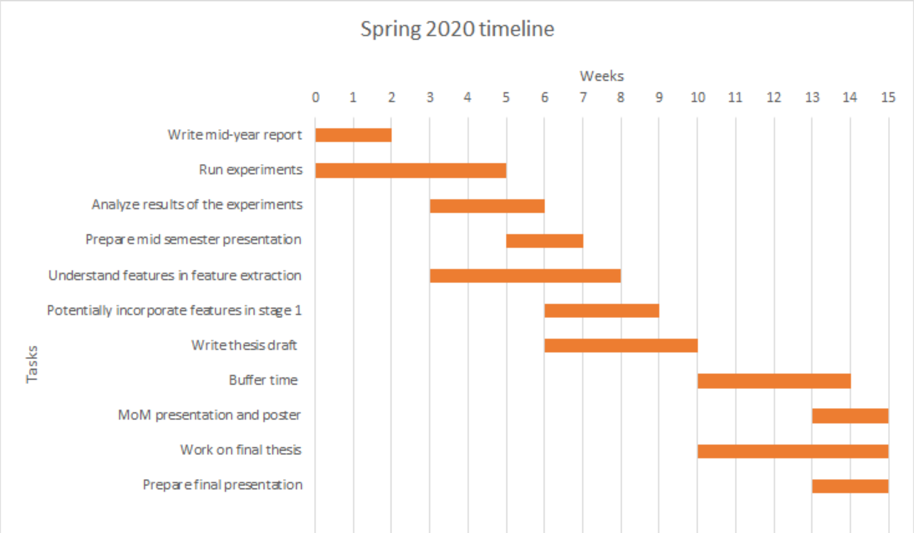
Predicting the size of the candidate set for effective retrieval:
Culpepper et al. [2] based their work on the fact that the final ranked list of documents is relatively insensitive to the quality of the initial candidate set in terms of early precision. They realized that if the candidate set has some good quality documents then the machine learning model in the candidate set re-ranking stage can identify and re-rank them high. Using this idea, they trained a cascade of binary random forest classifiers based on seventy pre-retrieval static features (e.g., term frequency, document frequency, etc.,). The classifier predicts the best value of the candidate set size (K) on a query-by-query basis, while minimizing the effectiveness loss and maximizing the efficiency gain. This idea of predicting optimal K on per query basis is beneficial because it allows the candidate set generation stage to potentially retrieve fewer documents. As a byproduct, this will decrease the cost of extracting document features in the feature extraction stage, thus, serving in maximizing efficiency. The binary cascaded classification is able to achieve up to a 50 % reduction in average candidate set size K.

Quality vs efficiency in learning-to-rank models:
Capannini et al. [3] investigated the training parameters of different machine learning algorithms and their impact on the quality versus efficiency tradeoff. Experiments concluded that there is no overall best algorithm, but the optimal choice depends on the time budget available for the multi-stage retrieval system. More precisely, the paper analyzes four broad families of learning-to-rank algorithms, namely: (1) *linear combinations*, (2) *artificial neural networks*, (3) *forests of regression trees*, and (4) *forests of oblivious trees.* The performance of each algorithm was plotted on a quality-cost curve, where quality was measured using NDCG (standard quality measure in IR) and cost was quantified as the time taken to score the documents. Using these quality-cost curves, authors were able to recommend the best algorithm for a given time budget. The paper also suggested that for a maximum time budget of 100 microseconds, the LTR algorithms lambda-mart and GBRT are most effective.

Early exit optimizations for additive machine learning:
Combazoglu et al. [4] proposed four different optimization strategies that allow short-circuiting score computation in additive machine learning systems. The optimization strategies are: (1) *early exit using score threshold* (EST), (2) *early exit using capacity threshold* (ECT), (3) *early exit using rank threshold* (ERT), and (4) *early exit using proximity threshold* (EPT). In EST, exits are based on comparisons between accumulated scores and offline-computed score thresholds. In ECT, exit decisions are based on a partial set of previously seen document scores. This is done by maintaining a maximum score heap. Afterwards, if the accumulated score of a document is less than the minimum score in the heap, the document score computation is short circuited. Early exits decisions in ERT are based on comparisons between the current ranks of documents (computed over all documents) and offline-computed rank thresholds. EPT utilizes the idea of scoring the documents that have scores close to the score of the document at the K^{th} rank. These strategies are able to expedite the score computations by more than four times with almost no loss in result quality [4].

Timeline


Fall 2019 timeline
Weeks
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Tasks
Literature review
Write Abstract
Identify frameworks for different stages
Write prospectus for senior thesis
Make mid-sem presentation
Set up framework for stage 1
Set up framework for stage 2
Set up framework for stage 3
Link stage 1 and stage 2
Link stage 2 and stage 3
Set up data for experiments
Design experiments
Run experiments on data
Make public poster presentation


Spring 2020 timeline
Weeks
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Tasks
Write mid-year report
Run experiments
Analyze results of the experiments
Prepare mid semester presentation
Understand features in feature extraction
Potentially incorporate features in stage 1
Write thesis draft
Buffer time
MoM presentation and poster
Work on final thesis
Prepare final presentation

Bibliography

[1] N. Asadi and J. Lin. Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In Proc. SIGIR, pages 997–1000, 2013.

[2] J. Shane Culpepper, Charles L. A. Clarke, and Jimmy Lin. 2016. Dynamic Cutoff Prediction in Multi-Stage Retrieval Systems. In Proceedings of the 21st Australasian Document Computing Symposium (ADCS '16), Sarvnaz Karimi and Mark Carman (Eds.).

[3] B. B. Cambazoglu, H. Zaragoza, O. Chapelle, J. Chen, C. Liao, Z. Zheng, and J. Degenhardt. Early exit optimizations for additive machine learned ranking systems. In Proc. WSDM, pages 411–420, 2010.

[4] Capannini, Gabriele & Lucchese, Claudio & Nardini, Franco Maria & Orlando, Salvatore & Perego, Raffaele & Tonellotto, Nicola. (2016). Quality versus efficiency in document scoring with learning-to-rank models. Information Processing & Management. 52.10.1016/j.ipm.2016.05.004.

[5] Matthias Petri, J. Shane Culpepper, and Alistair Moffat. 2013. Exploring the magic of WAND. In Proceedings of the 18th Australasian Document Computing Symposium (ADCS '13). ACM, New York, NY, USA, 58-65.

[6] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. Found. Trends Inf. Retr. 3, 4 (April 2009), 333-389.

[7] Qiang Wu, Christopher J. Burges, Krysta M. Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. Inf. Retr. 13, 3 (June 2010), 254-270.

[8] Friedman, J.H. (2001). Greedy function approximation: a grasient boosting machine, Annals of Statistics, 1189-1232.

[9] Macdonald, C., R. L. Santos, I. Ounis, and B. He. 2013b. "About Learning Models with Multiple Query-dependent Features". ACM Trans. Inf. Syst. 31(3): 11:1–11:39. issn: 1046-8188. doi: 10.1145/ 2493175.2493176.