QATAR UNIVERSITY

College of Engineering

Department of Computer Science and Engineering

Query Performance Prediction for Microblog Search

Maram Ghanem Hasanain

This thesis is submitted to Qatar University in partial
fulfillment of the requirements for the degree of
Master of Science in Computing

September 2014

Declaration

To the best of my knowledge, this thesis contains no material previously published or written by another person or institution, except where due reference is made in the text of the thesis. This thesis contains no material which has been accepted for the award of any other degree in any university or other institution.

Name Maram Ghanem Hasanain

Signature _____ Date _____

Committee

The thesis of **Maram Ghanem Hasanain** was reviewed and approved by the following:

We, the committee members listed below, accept and approve the Thesis/Dissertation of the student named above. To the best of this committees knowledge, the Thesis/Dissertation conforms the requirements of Qatar University, and we endorse this Thesis for examination.

Memebers:

Dr. Abdelkarim Erradi

Prof. Ali Jaoua

Prof. Franciska de Jong

Supervisor:

Dr. Tamer Elsayed

Signature _____ Date _____

Abstract

Microblogging has recently become an integral part of the daily life of millions of people around the world. With a continuous flood of posts, microblogging services (e.g., Twitter) have to *effectively* handle millions of user queries that aim to search and follow recent developments of news or events. A microblog search system can benefit from techniques of *query performance prediction* (QPP) to enhance effectiveness of microblog search. Predicting the effectiveness of retrieval of search queries was extensively studied in domains such as the Web and news. Yet, the different nature of data and search task in microblogs triggers the need for re-examining the problem in this context.

In this thesis, we studied the performance of 37 state-of-the-art query performance predictors in the context of microblog ad-hoc search using the two most-widely used tweets collections: Tweets2011 and Tweets2013. We also proposed several variants to some of the existing predictors to help improve prediction quality. In total, we experiment with 73 predictors (37 existing and 36 proposed). We evaluated quality of prediction of retrieval effectiveness using four retrieval models that were used in microblog search. We also studied prediction quality of predicting average precision (AP) at cut-off 1000 and precision at rank 30 (P@30) that are retrieval effectiveness measures commonly-used in evaluating microblog search. Finally, we worked on combining predictors using linear regression to study whether we can improve prediction with a combined prediction model.

We conducted several experiments to help answer a list of research questions. The results showed that query performance prediction in microblog search is reasonably effective and can be further improved. The variants we proposed were generally the best in predicting AP and P@30 across different retrieval models and with different collections. Furthermore, combining predictors achieved up to 20% improvement over individual predictors with some retrieval models. These promising conclusions promote the need for further work on QPP in the context of microblog search. The results also encourage utilizing QPP in applications that can help improve effectiveness of microblog search.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I thank my family for being there for me at all times. I thank my mother Amnah and my father Ghanem, for their continuous love, support, and prayers that helped make this thesis a reality. I thank my youngest brothers, Anas and Mo'men, for all the joyful moments they made sure I have in my darkest times, and for all the chocolate and laughs they have given me. I thank my older brother Bara', for being a good example to me, teaching me persistence, and showing me that hard work pays off. I would also like to thank my sisters Maiss and Rawan with whom I shared my life and bedroom; sorry for keeping you awake while working late at night!

My deepest gratitude goes to my supervisor Dr. Tamer Elsayed, who taught me how to conduct high-quality research. His selfless contribution of time and guidance helped shape the researcher and person I am today. I appreciate his insightful critiques, days he spent reviewing this work, and all invaluable comments and advice. I deeply thank him for believing in me, even when my enthusiasm and hope were fading away. I am really grateful to have such caring, fun, and humble mentor with whom I share a great vision and big goals.

I would like to thank my friends for their encouragement and the great discussions we had. They helped me learn to appreciate my life and the work I am doing. A special thank you goes to my great friend and sister Tara who has put up with me for three years now! She was always willing to listen to me mumbling about my work, and allowed me to think out loud, even when she had no idea what I was talking about.

This work was made possible by NPRP grant# NPRP 6-1377-1-257 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the author.

Dedication

To my brothers and sisters in Gaza. For all your sleepless nights and terror you withstood for our rightful cause. Your endless courage and persistence illuminated my heart and mind, and made me pursue my journey more determined than ever before.

Chapter 1

Introduction

In recent years, users have turned to microblogging services, such as Twitter[1], to share information that is as personal as their daily life details, up to the most general topics discussed world-wide. Twitter is indeed one of the fastest growing microblogging services with more than half a billion tweets posted daily.[2] Twitter users usually share information, news, and opinions about ongoing events via *tweets*, where a tweet is the unit of information sharing on Twitter, and can have a maximum of 140-characters. A tweet can include textual content that can possibly contain special elements like hashtags[3] and mentions.[4] A tweet can also contain one or more URLs pointing to webpages. Due to the vast growth of Twitter use and to the availability of Tweets test collections [44, 35], we focus our discussion on microblog search to tweets search in this thesis.

With the large number of tweets posted daily, a large number of queries are being issued by Twitter users, who of course as any typical users posting queries, are expecting relevant tweets to their queries. In a typical microblog ad-hoc search system, a user poses a query reflecting her information need and the system responses by a set of retrieved microblogs usually arranged in a form of a ranked result list where tweets are ranked in a descending order by their potential relevance (that is usually determined by a retrieval score assigned by the system and is hidden from the user). The result list includes supposedly relevant microblogs to that information need. Some users queries might be handled *effectively* by the system, i.e., the system will manage to retrieve relevant microblogs and rank them high in the result list for those queries. However, other queries can be difficult for the system to answer resulting in a poor quality of results.

The microblog search system can attempt to improve retrieval effectiveness for poorly-performing queries. Yet, for the system to do so, it should be able to accurately-enough *estimate* (or *predict*) how satisfied the user will be with the retrieved results prior to presenting the result list to the user, and specifically in the general situation where

[1] https://twitter.com/

[2] https://blog.twitter.com/2013/celebrating-twitter7

[3] A hashtag is constructed using the # symbol followed by one or more words. Hashtags are usually added to a tweet to reflect the tweet's topic.

[4] A mention is represented by the @ symbol followed by a Twitter username. Mentions are means of *tagging* other Twitter users in ones tweet for different reasons. Replying to a tweet posted by a user will result in a mention of the original user to appear in the reply tweet. Replies are among the most common reasons that mentions appear in tweets.

the system lacks user-provided relevance information. The process of doing so is called **Query Performance Prediction**.

1.1 Query Performance Prediction (QPP)

Query Performance Prediction (QPP) is the problem of predicting retrieval performance for a query given: a) the query, b) a retrieval model and c) a collection of documents. Prediction is usually carried in absence of relevance information [62, 3]. We will be referring to the approach that performs prediction given these variables by a *predictor* that will eventually result in a *predicted quality* of the results returned by the system given the query.

A large number of research studies have studied methods of query performance prediction, ranging from methods that only examine the query terms (i.e., *pre-retrieval* predictors) [23, 61, 20], to methods heavily relying on analyzing the retrieved documents for that query (i.e., *post-retrieval* predictors) [6, 25, 63, 9, 54]. Most of these studies were conducted on ad-hoc search in the news and Web domains [6, 25, 63, 54]. Few research studies studied this problem in the context of microblog search [18, 50].

1.2 Motivation

Recent studies have pointed out the high temporality in information dissemination in Twitter where a large portion of tweets are revolving around breaking news and events [30, 57]. The work of Teevan et al. [57] studied the querying behavior in Twitter and found that Twitter users post queries that are temporal. Work of Kwak et al. [30] also showed that Twitter users tend to be interested in retrieving relevant and *fresh* tweets. This high manifestation of temporal aspects in tweets and queries, in addition to the enormous number of tweets posted in Twitter, demands effective and real-time response to user queries. The problem of achieving effective retrieval is particularly severe in Twitter (and microblogs in general) due to the short length of queries used in microblog search (~2 words on average [57]) and the short length of tweets.

Finding effective ways to improve retrieval can increase user's satisfaction in the retrieved microblogs for a given query. A microblog search system can attempt to improve performance of *all* queries issued to the system using methods like query expansion, but such methods can actually result in degraded retrieval effectiveness for some queries [1, 7]. Moreover, with the enormous number of queries issued to microblog search systems, attempting to improve retrieval performance for all queries can be time consuming hindering real-time response expected in such temporal medium. Query performance prediction can support more informed decisions when it comes to handling queries submitted to a microblog search system. If performed efficiently and effectively-enough in the context of microblog search, QPP can allow for higher flexibility in terms of which queries to handle differently and possibly, selectivity in methods to use to improve retrieval effectiveness for a query.

Previous studies have focused on QPP in retrieval tasks in the context of typical TREC[5] Web and news collections. Web and news documents are generally long, well-formed, and non-conversational. The very short microblogs are naturally different from these documents as they tend to be temporal, conversational and very informal. The distinct features of the microblog search task and the data triggers the need to revisit the problem of query performance prediction in such domain.

1.3 Research Questions

In this thesis, we target six main research questions.

RQ1: How well do the state-of-the-art predictors perform in the context of microblog search?

A large number of predictors have been proposed in different domains. No previous studies have reported *comprehensive* results on the quality of these predictors in microblog search. Therefore, we are interested in examining the performance of these state-of-the-art predictors in microblog search to establish a baseline for QPP in this context. We focus on examining best performing predictors in different prediction scenarios, considering predictors that are:

- Non-microblog-specific: predictors originally designed for retrieval contexts other than microblog [6, 23, 25, 63, 61, 20, 9, 54].
- Microblog-specific: predictors originally proposed in the context of microblog search [50].

RQ2: Can we improve QPP in the context of microblog search?

Most of existing work has studied QPP in domains such as news and Web [6, 25, 63, 54]. Little work exist on this problem in the context of microblog search [18, 50]. Thus, we argue that current state-of-the-art prediction can be further improved with specific focus on the context of microblog search.

RQ3: Will the predictors' performance be consistent across different retrieval models that are used in microblog search?

As has been discussed in Section 1.1, QPP is usually dependent on the retrieval model used. In typical QPP literature, the query-likelihood model [46] is among the most common models considered with prediction [6, 63, 3, 54]. In microblog search, high temporality of the task and the data resulted in a large focus on using *temporal* retrieval models for search [13, 12, 15, 36, 14]. Moreover, the very short length of microblogs and queries turned attention to utilizing context expansion methods in microblog retrieval [4, 15, 58, 42]. Since such models have shown effectiveness in microblog search, we argue

[5] http://trec.nist.gov/. TREC refers to the Text REtrieveal Conference that is a major evaluation conference held yearly. Along with the conference, several test collections are released for evaluation purposes of retrieval tasks, covering a wide-spectrum of types of documents including tweets.

that a study of QPP in this context should consider such models. Thus, we examine QPP considering different types of retrieval models usually used in microblog search including:

- Standard QL model
- Temporal models
- Query expansion-based ones

Our aim with this question is to study whether prediction will be robust across different retrieval models used in this context.

RQ4: Will their performance be consistent across different test collections?

Prediction is performed given a test collection. A test collection is composed of a document collection with an associated set of queries, and relevance judgments for those queries indicating what documents in the collection are relevant to each [51]. Therefore, the test collection considered in prediction has its influence on prediction quality.

In this thesis, we examine two tweets collections: Tweets2011 [44], and Tweets2013 [35]. The first contains around 16 million tweets with 108 queries and the second has around 243 million tweets with 60 queries. One might think that these collections are indifferent and thus prediction quality should be consistent across them. Yet, as will be discussed in Section 4.1.1, a closer look to these two collections shows that they are different in several aspects. And since these two collections are the commonly-used ones in the state-of-the-art microblog search studies, we work on examining the consistency of performance of predictors across them.

RQ5: Will predictors generalize to different retrieval performance measures?

In retrieval evaluation, the actual performance of a query can be computed using one or more *retrieval effectiveness measures*, computed given the retrieved result list and the relevance judgments for that query [51]. In microblog search, average precision at cut-off 1000 (AP) and precision at rank 30 (P@30) are among the most-commonly used retrieval evaluation measures [44, 4, 58, 55, 35, 42]. In prediction, a predictor is used to predict the performance of a query by predicting one of these retrieval effectiveness measures. In typical QPP literature, prediction aims at predicting AP of retrieval [6, 25, 63, 3, 9, 54]. Due to the common use of P@30 in microblog search evaluation, we study the prediction quality of P@30, in addition to AP.

RQ6: Can we improve prediction quality in the studied domain using a combination of predictors?

Using individual predictors can be an effective prediction paradigm. However, combining predictors resulted in enhanced prediction quality over individual predictors in different domains [60, 10, 25, 63, 20, 53, 18, 50]. Thus, we work on combining predictors using linear regression in an attempt to improve prediction quality in this context.

1.4 Contributions

Given the previously discussed research questions, our contributions in this work are summarized as follows:

1. This is the first *extensive* study of query performance prediction in the context of microblog search. The study's distinction from earlier state-of-the-art work in this context is prevalent in its following characteristics:
    - We examine a total of 37 pre- and post-retrieval predictors, including predictors proposed in general contexts (such as news and Web search), and predictors proposed in the context of microblog search.
    - Experiments are carried with the two most-widely used microblog test collections (Tweets2011 [44, 55] and Tweets2013 [35]), with around 170 queries in total.
    - In prediction, we consider four different retrieval models. These models are representative of important types of models usually used in this context including: typical query-document similarity-based, temporal, query expansion and temporal query modeling retrieval models.
    - We conduct extensive experiments on the quality of prediction of P@30 as one retrieval effectiveness measure usually used in this context, in addition to predicting average precision which is the typical measure considered in QPP literature [6, 25, 63, 3, 9, 54].
    - The quality of prediction in such context is studied with focus on both individual predictors and combinations of them using linear regression.
2. In this work, we improve the quality of state-of-the-art prediction in the context of microblog search using several ways, including:
    - Proposing several prediction variants to some of the existing predictors which significantly improved prediction within different microblog search scenarios. Moreover, we believe that some of these variants are general enough and can help improve prediction quality in other contexts such as news or Web search.
    - Combining predictors using linear regression to produce a prediction model that is generally significantly better than individual predictors.
3. Our study provides a strong baseline for query performance prediction in the context of microblog search, that can be further improved or used in applications to support retrieval in microblogs.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. We first present a general background on QPP and a comprehensive review of related work in Chapter 2. Chapter 3 describes the query performance predictors we used in this study, in addition to the proposed variants

of predictors and the retrieval models used. Experimental setup is then presented and comprehensive results are discussed in Chapter 4, followed by the conclusion and some guidelines for future work in Chapter 5.

Chapter 2

Background and Related Work

In this chapter, we provide some background information on QPP and discuss different existing predictors. We present some of the applications in which QPP can be used. We then highlight aspects of different retrieval tasks in which QPP has been studied. We also discuss the existing work on QPP for microblog search.

2.1 Query Performance Prediction (QPP)

Query performance prediction can provide useful information to better guide several retrieval tasks; this is why it has been widely investigated in literature. As discussed in Chapter 1, query performance can be estimated using what we call a *predictor*. Predictors can be categorized into two main categories, *pre-* and *post*-retrieval predictors [3]. In the coming two sections, we discuss each of these categories.

2.1.1 Pre-retrieval Predictors

As illustrated in Figure 2.1, pre-retrieval predictors are those predictors computed *prior* to the retrieval step, based on analyzing the query expression and the statistics of the query terms in the collection [23, 21, 61, 3]. In general, pre-retrieval predictors can be efficiently computed since they rely on the query terms and collection statistics which are usually available at indexing time. Efficiency in prediction is a desirable feature when using prediction in some information retrieval (IR) tasks where quick system response is needed.

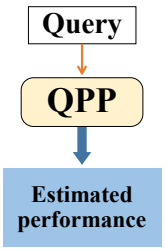


Figure 2.1: An illustration of the process of *pre*-retrieval prediction.

Pre-retrieval predictors can be divided into linguistic and statistical predictors [3]. In this work, we focus on statistical predictors that depend on the query terms distribution in the collection assuming that this distribution affects the retrieval performance.

One category of these predictors are those based on the inverse document frequency (idf) [39] of query terms [45, 23]. To compute the values of these predictors, idf of each query term is first computed, then the predicted values are computed as the maximum (**MaxIdf**), sum (**SumIdf**), average (**AvgIdf**), variance (**VarIdf**), and standard deviation (**DevIdf**) of the idf values over all query terms. Inverse collection term frequency ($ictf$) [31] is another query term statistic that can also be used as a basis to compute predictors in a similar way [45, 23]. Both idf and $ictf$ are used as measures of a term *rareness* in the collection. Predictors based on these measures assume that a query with infrequent terms is easier to answer. These predictors have demonstrated notable correlation with retrieval effectiveness over different collections [21].

Following the same approach considering per query term score, another category of predictors is based on computing a score for collection-query similarity (SCQ) [61]. The predicted values are the maximum (**MaxSCQ**), sum (**SumSCQ**), average (**AvgSCQ**), etc. of the SCQ values over all query terms. Predictors based on SCQ have shown some correlation to query difficulty with some collections [21, 54].

He and Ounis [23] proposed two predictors that can be used as indicators of the specificity of the query. Contrary to the aforementioned predictors, both of these predictors are computed based on an analysis over all query terms at once. The simplified clarity score (SCS) estimates the Kullback-Leibler divergence (KL-divergence) [28] between a language model of the query, constructed based on the query terms, and a language model of the collection [32]. A *language model* in this context, is a probabilistic distribution of terms composing a piece of text. Throughout this work, we assume term independence and construct a language model using unigrams of the text [46]. This predictor is a simplified version of a post-retrieval predictor called Clarity [6] that will be discussed in Section 2.1.2.

Query Scope (**QS**) [23] is another predictor proposed by He and Ounis as a measure of query-specificity that is estimated based on the proportion of documents containing at least one of the query terms to the total number of documents in the collection [23]. A high query scope indicates that a large set of documents will possibly be retrieved for a query making it difficult to discriminate documents that are actually relevant to the query. Both SCS and QS had notable prediction quality with some retrieval models and collections [23, 21].

Other pre-retrieval predictors exist, such as the predictors based on Pointwise Mutual Information (**PMI**) [19] that estimate query terms relatedness by measuring the probability of co-occurrence of each two query terms in the collection. These predictors assume that a query with highly-dependent terms is easy to answer. There are also predictors based on Term Weight Variability (**VAR**) [61] of each query term, that measures the variance of weights of a query term in all documents containing it. A higher overall variance of query terms weights indicate that the query is easier to answer since it is easier to discriminate relevant documents from irrelevant ones.

2.1.2 Post-retrieval Predictors

As Figure 2.2 shows, post-retrieval predictions are usually computed using a *result list* retrieved through a retrieval model given the query. This indeed adds an overhead to the prediction process. However, these predictors are usually better in reflecting the quality of retrieval compared to pre-retrieval predictors since they analyze the actual documents resulting from retrieval [3].

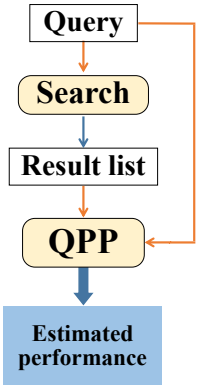


Figure 2.2: An illustration of the process of *post*-retrieval prediction.

Carmel and Yom-Tov [3] classified post-retrieval predictors into 1) clarity-based predictors, 2) robustness-based predictors, and 3) score distribution analysis-based predictors.

Clarity-based Predictors

Clarity-based predictors are based on estimating the *coherence* of the result list with respect to the collection [6]. The rationale behind these predictors is that an easy-to-answer query will have documents in the result list to be focused around the query topic [3]. More formally, a model induced from the documents in the result list, will be distinct from a model of the collection [54]. One of the widely-used clarity predictors is the **Clarity** score (**CLR**) [6] that is estimated by computing the KL-divergence between a language model induced from the result list and a language model of the collection.

Diaz and Jones proposed an analogy for the Clarity score but in the time domain [11]. The value of their predictor, ***t*-CLR**[1] [11, 25], is also computed by the the KL-divergence between the query and collection models, but using *temporal profiles* of the query and the collection instead of language models. A temporal profile for a query can be constructed using a probabilistic distribution of documents in the result list in the time domain given documents timestamps. Similarly, a temporal profile of the collection is constructed considering timestamps of all documents in the collection.

Other clarity-based predictors exist, like the weighted Clarity score proposed by Cronen-Townsend et al. [7]. This variant of Clarity introduces term weighting to the KL-divergence between the query and collection language models, where query terms are given a weight higher than others terms in the collection.

[1]The predictor was originally called *TemporalKL* but we refer to it as *t*-CLR to maintain a uniform naming convention of predictors throughout this work.

Robustness-based Predictors

Robustness predictors aim at estimating the robustness of the result list to perturbations in elements participating in the search task; a robust result list indicate an easy-to-answer query [3]. Query Feedback (**QF**) [63] is one of the robustness predictors considering query perturbations. This predictor measures the similarity between the original result list and a result list retrieved by inducing a new query from the original results. A high similarity indicates an easy query.

Score Distribution Analysis-based Predictors

These predictors work on analyzing the distribution of retrieval scores of documents in the result list [3]. Weighted information gain (**WIG**) [63] is one such measure. It estimates the divergence between the average retrieval score of documents in the result list and the collection retrieval score (considering the collection as one long document), assuming a higher divergence with high average retrieval score predict easier queries. This predictor was originally proposed considering Markov random field model of retrieval [41], but it was later reduced to a version that is based on query likelihood model [62].

Standard deviation of retrieval scores in the result list can be used as an indicator of query performance [52, 47, 9, 54]. A high standard deviation of scores indicate a lower query drift and hence an easy query [54]. The normalized standard deviation (**NSD**) [9] is one of these predictors. It is computed by normalizing the standard deviation of scores by the query length and considering an automatic approach to set the result list size. The normalized query commitment (**NQC**) predictor [52, 54] is also computed considering the standard deviation of retrieved documents scores but by normalizing it by a score of the collection (considering the collection as one long document) and using a fixed result list size with all queries.

2.2 QPP Applications

Several studies have gone beyond studying QPP as a goal to utilizing QPP in applications to enhance retrieval effectiveness. One line of research has focused on utilizing QPP for query expansion (QE) related applications. Most of these applications are motivated by the fact that QE is not always helpful in improving retrieval quality. More specifically, (1) QE can sometimes result in degraded retrieval effectiveness of some queries but not others [1, 7], and (2) the way of expansion that should be applied to one query to improve its performance might not be the same for other queries [24, 38]. To handle the first issue, QPP has been used to perform *selective* QE where expansion is only applied to queries that are predicted to improve with expansion [1, 7]. QPP is sometimes used to perform *dynamic* QE to solve the second problem, where QPP helped adaptively set the level of expansion [38], or the source collection to use in expansion [24], per query.

QPP was also used in building an adaptive query suggestion system [37]. Liu et al. [37] used performance predictors as features in a learning-to-rank approach to learn to rank candidate suggestions for a query. They also used predictors as features in a classifier that adaptively selects the query suggestion approach to use based on the difficulty of the query. Jones and Diaz employed performance predictors as features but for classification.

Their work aimed at using *temporal* performance predictors as features to characterize the temporal nature of queries. Using supervised machine learning techniques, these features were used in automatic classification of queries into different temporal types [25]. Performance predictors were again used as features in a learning-to-rank approach, but this time in a system for query length reduction [29]. Kumaran and Carvalho [29] used predictors such as Clarity and SCS as features in learning-to-rank sub-queries of an originally long query based on their predicted retrieval effectiveness. Based on this approach, the system selects the sub-query with the highest predicted performance to be used instead of the original query.

2.3 QPP in Retrieval

Generally, existing work on QPP has mainly targeted ad-hoc search [6, 63, 20, 9, 54, 56]. Yet, QPP has been investigated in other retrieval tasks. Cronen-Townsend et al., and Krikon et al. studied QPP in the context of passage retrieval for question answering [8, 27]. The work of Cronen-Townsend et al. [8] uses the Clarity score computed considering the question and the answer passages retrieved for it as the query and documents in typical ad-hoc search. Their work has shown that there is a weak correlation between the passages Clarity score and the performance of the question answering system indicating that prediction can possibly be used for question difficulty prediction.

The recent work of Raviv et al. [49] use QPP in the context of entity retrieval. In that work, a retrieved entity for a short query is represented by three representations: the entity document represented by the Wikipedia page of the entity, entity type, and entity retrieval score given to it by the retrieval system. In their approach, they adapt existing predictors such as the Clarity score, WIG, NQC, and NSD to utilize different representations of the entity in prediction. They also propose an entity clustering-based predictor assuming that a retrieved entity list with entities that are similar and with high retrieval scores is probably effective.

Most of the previously mentioned studies focused on retrieval tasks using news and Web documents. For example, the first proposed predictor, Clarity [6] was mainly designed and tested for ad-hoc retrieval over typical TREC collections mainly including news articles. Experiments of both NSD [9] and NQC [54] were focused on Web and news search. WIG and QF were originally designed with focus on Web search [63]. And in the recent work [49] on prediction in entity retrieval, prediction was tested using Wikipedia pages.

Similar to most of the existing work, we are also investigating ad-hoc search in document collections, but the nature of documents we are handling is very different. Tweets (the microblogs we are using in our experiments) are very short (with a maximum of 140 characters), conversational in nature, and usually informal. Moreover, tweets are driven by high variety of use cases which introduces a lot of noise and heterogeneity.

Understanding the specific features of tweets in addition to the nature of topics and language used in Twitter is one of the main challenges to consider when working on QPP in the context of microblog search. In the coming section, we introduce a summary on existing work experimenting with state-of-the-art predictors in the domain of microblog

search. We also present a study that attempted QPP in the context of microblog search while capturing the specific nature of tweets in prediction.

2.3.1 QPP in Microblog Search

Up to our knowledge, our study [18] is the first extensive study of QPP in microblog search. It worked on experimenting with existing pre- and post-retrieval predictors on two tweets collection [44, 44] and using three retrieval models. The initial results showed that there is a considerable correlation between the predicted average precision by some of the predictors and the retrieval effectiveness. It also showed that a temporal predictor is probably the more suitable predictor for the microblog search task. The study also demonstrated that combining predictors using linear regression results in enhanced prediction quality which conforms with what have been reported in other domains [60, 10, 25, 63, 20, 53].

Parallel to the work of Hasanain et al. [18], Perez and Jose [50] have published their work on QPP in microblog search. The aim of their work was to study QPP in order to utilize it for selective automatic query expansion. Similar to this study, they have focused on studying the quality of prediction of average precision in addition to prediction quality of precision at rank k. However, they ran the predictors using the DFRee retrieval model [2] only where we study prediction across four models. Differently from the evaluation approach in this work, they combined queries associated to the two tweets collections, Tweets11 [44] and Tweets13 [35], into one set. Moreover, no parameters tuning was performed for the predictors, although some of these predictors can be notably sensitive to the parameters used in them (e.g. WIG and NQC are sensitive to the number of documents in the results list [3, 54]).

Perez and Jose [50] presented results of their experiments with some of the existing predictors, in addition to proposing a good number of post-retrieval predictors including predictors designed considering specific features of microblogs such as hashtags and URLs appearing in tweets. We briefly discuss their newly proposed predictors next.

Predictors based on documents Time Cohesion (TCH)

TCH [50] was proposed to measure the time cohesion of documents in the initially retrieved list, assuming that high cohesion indicates effective retrieval. This measure computes the difference in posting times of consecutive ranked documents in the list (given documents' timestamps). The predictors are the mean (**MeanTCH**), median (**MedTCH**), minimum (**MinTCH**), etc. of the TCH values over all documents.

Predictors based on query terms coverage (QTC)

Perez and Jose [50] proposed the QTC as a measure of query terms coverage by the documents in an initially retrieved list, assuming that a high coverage indicates an easier to answer query. QTC is computed per document in the list and the predictors values are the mean (**MeanQTC**), median (**MedQTC**), minimum (**MinQTC**), etc. of the QTC values over all documents.

Predictors based on top terms coverage (TTC)

With a similar intuition to QTC, TTC measures the documents coverage of the m most frequent terms in the retrieved list. TTC is computed per document and the predictors values are the mean (**MeanTTC**), median (**MedTTC**), minimum (**MinTTC**), etc. of the TTC values over all documents.

Hashtags ratio

The value of this predictor is the ratio of documents in the result list containing at least one hashtag.

URLs ratio

This predictor measures the ratio of documents in the result list containing at least one URL.

That study has showed that, in general, some of their newly proposed predictors have superior prediction quality compared to all of the existing predictors tested. The TTC-based predictors specifically have shown superior prediction quality of both average precision and precision at rank 10 compared to all other predictors.

In this work, we follow a similar approach to [18] and [50] to study prediction in microblog search, but examining prediction across four retrieval models, focusing on temporal and query expansion ones. We specifically study *idf*- and SCQ-based predictors, SCS, Clarity score (CLR), t-CLR, NQC, WIG, NSD, and all predictors proposed by Perez and Jose in [50]. We propose a variant to the Clarity score that uses temporal relevance modeling to construct the model of the query. We also propose variants to three categories of the post-retrieval predictors proposed by Perez and Jose [50] for microblog search.

Chapter 3

The Setup of the Study

To study query performance prediction in the context of microblog search, we experimented with a total of **37** existing pre- and post-retrieval predictors discussed in sections 3.1 and 3.2. In section 3.3, we propose some variants to a number of existing post-retrieval predictors in an attempt to improve QPP in the context of microblog search. We study the robustness of predictors across different retrieval models by experimenting with predictors across several retrieval models used in microblog search, presented in section 3.4. Section 3.5 presents our approach for combining predictors. By the end of this chapter, in section 3.6, we discuss some implementation issues considered while employing the study design described.

3.1 Pre-retrieval Predictors

To partially answer our first research question, we study the behavior of **9** existing pre-retrieval predictors in the context of microblog search. We basically experimented with two main categories of predictors. The first category of predictors, i.e., *idf*-based predictors, is based on the inverse document frequency (*idf*) [39] of query terms. *idf* is a measure of a term rareness in the collection. It can be computed as follows:

$$idf(w) = \log \frac{N}{df_w} \tag{3.1}$$

where N is the number of documents in the collection and df_w is the document frequency [39] of term w. Under this category, we considered the maximum (**MaxIdf**), sum (**SumIdf**), average (**AvgIdf**), variance (**VarIdf**), and standard deviation (**DevIdf**) of *idf* values of query terms. These predictors are designed to estimate the *specificity* of the query expression predicting that a query with uncommon terms is easier to answer [21, 3]

The other category, *SCQ*-based predictors, is based on a score for collection-query similarity (*SCQ*) [61]. The intuition behind this category of predictors is that a query that is more similar to the collection will be easier to answer. *SCQ* is defined as follows:

$$SCQ(w) = (1 + \log(cf_w)) \log \left(1 + \frac{N}{df_w}\right) \tag{3.2}$$

where cf_w is the collection frequency [39] of query term w in the document collection C. Under this category, we considered the maximum (**MaxSCQ**), sum (**SumSCQ**), and average (**AvgSCQ**) of SCQ values of query terms.

Additionally, we experimented with another query-specificity predictor that is the simplified clarity score (SCS) [23]. SCS estimates the KL-divergence between the query language model based on the query terms and the collection language model. A *language model* [46] is a probabilistic distributions of terms. Throughout this work, we assume terms independence and construct the language model using unigrams. The value of SCS is computed as follows:

$$SCS(Q) = \sum_{w \in Q} P(w|Q) \log \frac{P(w|Q)}{P(w|C)} \tag{3.3}$$

where $P(w|Q)$ is estimated using the maximum likelihood estimate (MLE) [39] as follows: $P(w|Q) = \frac{tf_{w,Q}}{|Q|}$, where $tf_{w,Q}$ is the term frequency of w in Q, $|Q|$ is the query length, and $P(w|C)$ is estimated by MLE over C.

In total, we worked with **9** pre-retrieval predictors since these predictors have generally had good prediction ability in different settings [19], and can be efficiently computed. Yet we gave more attention to post-retrieval predictors due to their superior prediction quality in general [3]. In the coming section, we present the post-retrieval predictors we studied.

3.2 Post-retrieval Predictors

We continue answering our first research question by studying the performance of **28** state-of-the-art post-retrieval predictors in the context of microblog search. Post-retrieval predictors require a list R of l retrieved documents in response to a given query Q, in order to predict the performance of Q [3]. As highlighted in Chapter 2, most of the existing predictors were designed and tested in Web and news search domains. In this section, we present **5** of these predictors selected based on their reported high prediction quality when experimented with different types of collections [25, 63, 9, 54]. We also present **23** newly proposed predictors [50] designed for microblog search. For each of the predictors presented next, the result list length l is a free parameter.

Clarity (CLR): CLR is one of the very first proposed predictors [6]. We chose to experiment with CLR since it has shown good prediction quality across different collections [62, 54]. In CLR, prediction is based on estimating the coherence of the list R with respect to the collection of documents C using the KL-divergence between the query language model induced by R and the collection language model. The query language model is represented as follows:

$$P(w|Q) = \sum_{D \in R} P(w|D)P(D|Q) \tag{3.4}$$

where $P(w|D)$ is estimated using the maximum likelihood estimate (MLE) [39] as follows: $P(w|D) = \frac{tf_{w,D}}{|D|}$, where $tf_{w,D}$ is the term frequency of w in D and $|D|$ is the document

length. Originally [6], linear smoothing with the collection model was used in constructing the document model $P(w|D)$, but we chose to use an unsmoothed model since this setting showed effective prediction quality in earlier studies [54]. $P(D|Q)$ is computed as the sum-normalized (over all documents in R) query likelihood of D [46]. Query likelihood is the likelihood that the document language model generated the query (details on this model can be found in section 3.4.1). $P(D|Q)$ can be computed as shown next:

$$P(D|Q) = \frac{\prod_{w \in Q} P(w|D)}{\sum_{D' \in R} \prod_{w \in Q} P(w|D')} \tag{3.5}$$

where $P(w|D)$ is computed by the MLE, smoothed using Dirichlet smoothing [39] with the collection language model. Finally, the clarity score is computed using KL-divergence as follows:

$$CLR(Q) = \sum_{w \in V} P(w|Q) \log \frac{P(w|Q)}{P(w|C)} \tag{3.6}$$

where V is the vocabulary set and $P(w|C)$ is estimated by MLE over C.

More recent studies have developed the predictors NQC, NSD, and WIG that work on analyzing the distribution of retrieval scores of documents in R. Studying these predictors is of interest to us because of their reported high prediction quality when experimented with different collections [63, 52, 9, 54] and because they can be computed more efficiently compared to other post-retrieval predictors such as the Clarity score. We describe these predictors next.

Normalized Query Commitment (NQC): NQC [54] measures the amount of query drift in the results list R; that is, the commitment of documents in R to aspects related to Q. The value of NQC is computed as follows:

$$NQC(Q) = \frac{\sigma_R}{|Score(Q, C)|} \tag{3.7}$$

where σ_R is the standard deviation of retrieval scores of documents in R. $Score(Q, C)$ is the retrieval score of the collection when viewed as one very long document, computed as follows:

$$Score(Q, C) = \sum_{w \in Q} \log P(w|C) \tag{3.8}$$

where $P(w|C)$ is estimated by MLE over C.

Normalized Standard Deviation (NSD): With a similar intuition to NQC, value of NSD [9] is computed as the standard deviation of document retrieval scores, but normalized by the square root of the query length instead of the collection score. It also differs from NQC, when computing the standard deviation, in considering only top documents in R with retrieval scores greater than $x\%$ of the score of the top-ranked document. The predicted value is computed as follows:

$$NSD(Q) = \frac{\sigma_{x\%}}{\sqrt{|Q|}} \tag{3.9}$$

where $|Q|$ is the query length and $\sigma_{x\%}$ is the standard deviation of retrieval scores of documents matching the $x\%$ cut-off criterion. x is a free parameter.

Weighted Information Gain (WIG): WIG [63] measures the difference between the average retrieval score of documents in R and the collection retrieval score. In this study, we adopted a reduced version that is based on query likelihood model [62], and thus the value of WIG is computed as follows:

$$WIG(Q) = \frac{1}{l} \sum_{D \in R} \frac{1}{\sqrt{|Q|}} \left(\sum_{w \in Q} \log P(w|D) - \sum_{w \in Q} \log P(w|C) \right) \tag{3.10}$$

where $P(w|C)$ and $P(w|D)$ are estimated by MLE over C and D respectively. We shorten the equation of WIG to be as follows:

$$WIG(Q) = \frac{1}{l} \sum_{D \in R} \frac{1}{\sqrt{|Q|}} (Score_{QL}(Q, D) - Score_{QL}(Q, C)) \tag{3.11}$$

where $Score_{QL}(Q, D)$ and $Score_{QL}(Q, C)$ corresponds to the query log-likelihood scores of D and C respectively present in equation 3.10.

Temporal Clarity (t-CLR): t-CLR [25] is a variant of the clarity predictor that emphasizes the *temporal* aspect of the data. We are interested in studying such predictor since temporal aspects of the data is important in the microblog search task [14]. t-CLR measures the KL-divergence between the *temporal profile* of the query (represented by $P(t|Q)$) and the *temporal profile* of the collection (represented by $P(t|C)$). A *Temporal Profile* is a distribution of documents under consideration in the time domain. We compute t-CLR as follows:

$$t\text{-}CLR(Q) = \sum_{t \in T} P(t|Q) \log \frac{P(t|Q)}{P(t|C)} \tag{3.12}$$

where T is the set of timestamps in the collection in unit of h hours and we consider h as a free parameter. For simplicity, in this work, we only consider timestamps in list R. $P(t|C)$ is estimated as a uniform distribution over all timestamps in C, and $P(t|Q)$ is estimated by first computing $\tilde{P}(t|Q)$ as follows:

$$\tilde{P}(t|Q) = \sum_{D \in R} P(t|D) P(D|Q) \tag{3.13}$$

where $P(t|D)$ is 1 for documents posted within the timestamp t, and 0 otherwise, and $P(D|Q)$ is estimated as in Equation 3.5. $P(t|Q)$ is then computed by smoothing $\tilde{P}(t|Q)$ with the collection temporal model as follows:

$$P(t|Q) = \lambda \tilde{P}(t|Q) + (1 - \lambda) P(t|C) \tag{3.14}$$

where the smoothing factor λ is another free parameter for this predictor.

A recent work [50] has proposed a set of new predictors that we also study. We give a lot of focus to these predictors since they show good prediction quality in the context of microblog search that is studied here. In addition, the recency of these predictors offers us room to propose possible enhancements to them. We discuss these next.

Query Terms Coverage (QTC): Perez and Jose [50] proposed using the QTC measure as a basis for a set of predictors to estimate how well the query is covered by documents in R, assuming a high coverage indicates high quality results. QTC is first computed *per document* as follows:

$$QTC(Q, D) = \frac{\sum_{w \in Q} minimum(tf_{w,D}, 1)}{|Q|} \tag{3.15}$$

Given this measure, the predictors are the mean (**MeanQTC**), median (**MedQTC**), minimum (**MinQTC**), maximum (**MaxQTC**), range (**RangeQTC**), upper quartile[1] (**UpQTC**), and lower quartile[2] (**LowQTC**) of the QTC values over all documents in R. SumQTC for example, can be computed as follows:

$$SumQTC(Q, R) = \sum_{D \in R} QTC(Q, D) \tag{3.16}$$

Top Terms Coverage (TTC): TTC was proposed by Perez and Jose [50] as another basis for prediction. This measure is used to estimate how well the m top occurring terms in R are covered by documents in R. To compute TTC, we first find the list L_m of most frequent m terms in R, where m is a free parameter. TTC can then be computed *per document* as follows:

$$TTC(L_m, D) = \frac{\sum_{w \in L_m} minimum(tf_{w,D}, 1)}{m} \tag{3.17}$$

Given this measure, the predictors are the mean (**MeanTTC**), median (**MedTTC**), minimum (**MinTTC**), maximum (**MaxTTC**), range (**RangeTTC**), upper quartile (**UpTTC**), and lower quartile (**LowTTC**) of the TTC values over all documents in R. For example, we compute SumTTC as follows:

$$SumTTC(L_m, R) = \sum_{D \in R} TTC(L_m, D) \tag{3.18}$$

Time Cohesion (TCH): Perez and Jose [50] proposed this *temporal* measure as a basis for prediction. TCH is used to measure *temporal cohesion* of documents in R assuming that high cohesion indicates that documents are discussing the same topic. TCH can be

[1] Computed as the median of the sub-list with the highest 50% of values from the list under consideration.

[2] Computed as the median of the sub-list with the lowest 50% of values from the list under consideration.

computed for each *pair* of consecutive ranked documents in R as follows:

$$TCH(D_i, D_{i+1}) = |t_i - t_{i+1}| \tag{3.19}$$

where t_i is the timestamp (in the unit of seconds) of a document D_i at rank i in the result list R. Given this measure, the predictors are the mean (**MeanTCH**), median (**MedTCH**), minimum (**MinTCH**), maximum (**MaxTCH**), range (**RangeTCH**), upper quartile (**UpTCH**), and lower quartile (**LowTCH**) of the TCH values over all documents in R. As an example, we show below how SumTCH is computed.

$$SumTCH(R) = \sum_{i=1}^{l-1} TCH(D_i, D_{i+1}) \tag{3.20}$$

Hashtags Ratio (#Rate): Perez and Jose [50] proposed a tweet-specific predictor that computes the ratio of documents in R containing at least one hashtag[3].

URls Ratio (UrlRate): Similarly to the #Rate, the UrlRate is the ratio of documents in R containing at least one URL [50].

In this work, we focused on studying these existing predictors in the context of microblog search. In addition, we propose variants of some of the existing predictors in an attempt to improve QPP in the context of microblog search.

3.3 Proposed variants of Existing Predictors

3.3.1 Exponential Time Cohesion (ExpTCH)

The time cohesion measure proposed in [50] as a basis for a set of predictors in the context of microblog search, considers linear differences between documents' timestamps. Recent work on microblog search [15] has shown that using an exponential temporal prior [34] in the query likelihood model [46] significantly improved retrieval effectiveness over using typical query likelihood model. This finding indicates that, generally, relevant tweets to a query are likely to better fit an exponential distribution over time. Starting by this intuition, we propose borrowing the exponential factor proposed in [34] to represent the time cohesion measure as follows:

$$ExpTCH(D_i, D_{i+1}) = e^{-r \cdot |t_i - t_{i+1}|} \tag{3.21}$$

where t_i is the timestamp (in the unit of days) of a document D_i at rank i in R. r is a decay rate parameter that should be tuned for this predictor. Given this measure, the predictors are then the mean (**MeanExpTCH**), median (**MedExpTCH**), minimum (**MinExpTCH**), maximum (**MaxExpTCH**), range (**RangeExpTCH**), upper quartile (**UpExpTCH**), and lower quartile (**LowExpTCH**) of the ExpTCH values over all

[3] A hashtag is represented by a # sign followed by one or more words (e.g., #Prediction). Hashtags are usually used as indicators of the microblog topic.

documents in R. For example, SumExpTCH is computed as follows:

$$SumExpTCH(R) = \sum_{i=1}^{l-1} ExpTCH(D_i, D_{i+1}) \tag{3.22}$$

3.3.2 Idf-based QTC

Predictors based on the QTC measure (discussed in section 3.2) consider the coverage of query terms in the retrieved documents [50]. In this measure, all query terms are considered equally helpful in representing the query topic, but actually some query terms might be more representative of the query topic compared to others. Thus, considering term weights in the QTC measure might help improve prediction quality by better capturing the query topic coverage in documents. Furthermore, we hypothesize that such weighting might be crucial for predicting using QTC in the context of microblog search due to the short length of the documents (i.e., tweets or microblogs). The very short microblog might not cover many terms of the query because of its short length and thus, weighting coverage considering background information from the collection can improve QTC's ability in representing the microblog query-coverage. We propose an *idf*-based term-weighted QTC measure computed as follows:

$$LIdfQTC(Q, D) = \frac{\sum_{w \in Q} idf(w) \cdot minimum(tf_{w,D}, 1)}{|Q|} \tag{3.23}$$

where we normalize the coverage by the query length $|Q|$ to avoid biased performance prediction due to the different lengths of the queries. Given this measure, the predictors are the mean (**MeanLIdfQTC**), median (**MedLIdfQTC**), minimum (**MinLIdfQTC**), maximum (**MaxLIdfQTC**), range (**RangeLIdfQTC**), upper quartile (**UpLIdfQTC**), and lower quartile (**LowLIdfQTC**) of the LIdfQTC values over all documents in R.

We also propose another *idf*-based variant to the QTC measure in which we normalize the *idf*-weighted coverage by the query specificity measured by the sum of *idf* values of all query terms. Such normalization can help in reducing bias in prediction since it allows normalized weighting for the covered terms in each document. We refer to this measure as IdfQTC computed as:

$$IdfQTC(Q, D) = \frac{\sum_{w \in Q} idf(w) \cdot minimum(tf_{w,D}, 1)}{\sum_{w' \in Q} idf(w')} \tag{3.24}$$

Given this measure, the predictors are the mean (**MeanIdfQTC**), median (**MedIdfQTC**), minimum (**MinIdfQTC**), maximum (**MaxIdfQTC**), range (**RangeIdfQTC**), upper quartile (**UpIdfQTC**), and lower quartile (**LowIdfQTC**) of the IdfQTC values over all documents in R.

3.3.3 Idf-based TTC

Following the same intuition used in LIdfQTC, we propose an *idf*-based term-weighted TTC measure computed as follows:

$$LIdfTTC(L_m, D) = \frac{\sum_{w \in L_m} idf(w) \cdot minimum(tf_{w,D}, 1)}{m} \tag{3.25}$$

Given this measure, the predictors are the mean (**MeanLIdfTTC**), median (**MedLIdfTTC**), minimum (**MinLIdfTTC**), maximum (**MaxLIdfTTC**), range (**RangeLIdfTTC**), upper quartile (**UpLIdfTTC**), and lower quartile (**LowLIdfTTC**) of the IdfTTC values over all documents in R.

We also propose normalizing the coverage of top terms by the overall terms specificity as follows:

$$IdfTTC(L_m, D) = \frac{\sum_{w \in L_m} idf(w) \cdot minimum(tf_{w,D}, 1)}{\sum_{w' \in L_m} idf(w')} \tag{3.26}$$

Given this measure, the predictors are the mean (**MeanIdfTTC**), median (**MedIdfTTC**), minimum (**MinIdfTTC**), maximum (**MaxIdfTTC**), range (**RangeIdfTTC**), upper quartile (**UpIdfTTC**), and lower quartile (**LowIdfTTC**) of the IdfTTC values over all documents in R.

3.3.4 Temporal Relevance Modeling-based Clarity (*trm*-CLR)

Due to the temporal nature of tweets, we propose using a temporal relevance model [26] as the query model in the Clarity score. The query model can be computed as follows:

$$P(w|Q) = \sum_{t \in T} P(w|t, Q)P(t|Q) \tag{3.27}$$

where t is a timestamp in unit of h hours and T is the set of timestamps in the collection. For simplicity, in this work, we only consider timestamps in list R in computing this probability. The parameter h is a free parameter for this predictor. $P(t|Q)$ is estimated as the normalized sum of retrieval scores of documents in R posted within t. The probability $P(w|t, Q)$ can be computed as follows:

$$P(w|t, Q) = \sum_{D \in t} P(w|D)P(D|t, Q) \tag{3.28}$$

$P(D|t, Q)$ is assumed to be uniform over all documents in R posted within t. $P(w|D)$ is computed using the MLE over D.

In addition to studying the performance of a large set of predictors in the context of microblog search, studying the robustness of predictors across different models is one of the objectives of this work. We are specifically focusing on temporal and query expansion models that are used in microblog search. In the following section, we present the four models we considered.

3.4 Retrieval Models

To examine the robustness of predictors across different retrieval approaches, we measured the quality of prediction with four different retrieval models that are representative of the types of models used for microblog search.

3.4.1 Query Likelihood (QL)

The Query Likelihood (QL) model [46] is typically used in related QPP studies [6, 25, 54]. In this model, documents can be ranked by the likelihood that their language models generated the query as follows:

$$P(D|Q) \propto P(Q|D)P(D) \tag{3.29}$$

Assuming a uniform document prior $P(D)$ and terms independence, documents can be ranked by

$$P(D|Q) \propto P(Q|D) = \prod_{w \in Q} P(w|D) \tag{3.30}$$

where $P(w|D)$ is initially computed using the MLE over D. To overcome the zero-probability problem, we further smooth the model of $P(w|D)$ using Dirichlet smoothing as follows:

$$P(w|D) = \frac{tf_{w,D} + \mu P(w|C)}{|D| + \mu} \tag{3.31}$$

where $P(w|C)$ is estimated by MLE over C and μ is a free parameter for this retrieval model.

3.4.2 PRF-based Query Expansion (QE)

Using Pseudo Relevance Feedback (PRF) [33] for query expansion has demonstrated good retrieval effectiveness in microblog search [40, 4, 43]. The typical PRF-based query expansion model expands a query using m terms extracted from the top k documents in an initially retrieved list R given Q. In here, we use a tf-idf [39] like measure to score terms over all documents in R as follows:

$$Score(w, R_k) = tf_{w,R_k} \cdot idf(w) \tag{3.32}$$

where R_k is the subset of R containing the top k documents. tf_{w,R_k} is the sum of term frequencies of w over all documents in list R_k and $idf(w)$ is computed as in equation 3.1. Once we expand the query with m terms, we use the query likelihood model to retrieve the final list of documents using the expanded query. Both m and k are free parameters fro this model.

3.4.3 Time-based Exponential Priors (t-EXP)

The t-EXP model [34] has shown good retrieval performance for recency queries in microblog search [13]. The model simply extends the QL model using an exponential decay

factor as a document prior as follows:

$$P(D|Q) \propto P(Q|D) \cdot r \cdot \mathrm{e}^{-r.t_d} \tag{3.33}$$

where $P(Q|D)$ is the query likelihood of the document D, r is a decay rate parameter, and t_d is the time difference in days between the posting time of D and the posting time of Q.

3.4.4 Time-based Query Relevance Modeling (t-QRM)

t-QRM [26] is a variant of the typical query relevance modeling approach [33] in which the relevance model of the query is temporal and computed as follows:

$$P(w|Q) = \sum_{t \in T} P(w|t, Q)P(t|Q) \tag{3.34}$$

where t is a timestamp in unit of days and T is the set of timestamps in the collection. For simplicity, in this work, we only consider timestamps in an initially retrieved list R_k (retrieved using the standard QL model) in computing this probability. $P(t|Q)$ is estimated as the normalized sum of retrieval scores of documents in R_k posted within t. The probability $P(w|t, Q)$ can be computed as follows:

$$P(w|t, Q) = \sum_{D \in t} P(w|D)P(D|t, Q) \tag{3.35}$$

$P(D|t, Q)$ is assumed to be uniform over all documents in R_k posted within t. $P(w|D)$ is computed using the MLE over D. We choose to model the final query using the m terms in R_k with the highest probability $P(w|Q)$. The QL model is then used to rank documents using this query model. Both the initial list size k and the number of terms in query model m are free parameters for this model.

3.5 Combining Predictors

Earlier work on combining predictors in different domains have reported noticeable improvements in prediction quality [11, 60, 25, 63, 53]. Studies examining combining predictors in microblog search domain have also reported similar observations [18, 50]. One of the research questions in this work is to study the effect of combining predictors on the quality of prediction. To answer this research question and given the promising results reported in literature, we also work on combining predictors.

To combine predictors, we employ linear regression to learn a combination model. In this approach, we construct a *linear* relationship between the vector of values of predictors and the overall predicted average precision for a given query [25, 3]. To learn such model, we train the model as follows:

$$AP(Q_i) = \overline{y}_i \overline{\beta} + \epsilon_i \tag{3.36}$$

where $i = 1, ..., n$ and n is the number of queries associated with known average precision values (computed using the actual relevance judgments) used in training the model.

$AP(Q_i)$ is the the average precision for a query Q_i, $\overline{y}_i$ is the vector of average precision values predicted by the set of predictors for Q_i, $\overline{\beta}$ is the vector of weights of the predictors, and ϵ_i is an error value. The goal of the training phase, is to *learn* the vector $\overline{\beta}$ that best fits the n equations of training queries by minimizing the root mean square error (RMSE). RMSE is represented as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(AP(Q_i) - \overline{y}_i\overline{\beta}\right)^2} \tag{3.37}$$

Different learning approaches can be used to learn the combination model. Cross Validation (CV) and Leave-one-out (LOO) are among the commonly used approaches in QPP literature [11, 25, 63, 20]. We discuss the specific training and testing approach used in the evaluation chapter.

Having laid down the theoretical basis of this study, we present next some specific details on how we have employed the previously discussed concepts to study QPP in microblog search.

3.6 Implementation Issues

When ranking documents using the QL, we use an effective and well-known implementation of query likelihood that is the log-likelihood defined as follows:

$$P(D|Q) \propto \log P(Q|D) = \sum_{w \in Q} \log P(w|D) \tag{3.38}$$

where $P(w|D)$ is computed as shown in equation 3.31. It should be noted that we use the ***natural logarithm*** in this equation and in all the aforementioned equations relying on logarithms.

The main implementation issue that we needed to handle is adapting *some* of the predictors to work across retrieval models other than the QL model. In Table 3.1 below, we show how we adapted these predictors to allow them to work with models other than the QL model.

Considering we worked with two query expansion models, we decided to use the *expanded queries* generated by these models instead of the original queries when computing post-retrieval predictors relying on query terms. We believe this will help improve prediction since the final result list returned by the system is actually based on the expanded query; an expanded query is assumed to be more related to the results compared to the unexpanded version. This indicates that the expanded query should be considered in prediction that is based on both the query and the result list. Experimental results supporting this conclusion can be found in Chapter 4. Yet, this approach comes with the caveat that prediction should be done online incorporated with retrieval to acquire the expanded queries for a specific query expansion model. Prediction can also be done offline, but the prediction system should (somehow) have access to the expanded queries generated by the exact query expansion model used with prediction.

Table 3.1: Adaptations introduced to some predictors allowing them to work with different retrieval models.

Predictor	**Adaptation**
CLR & t-CLR	The sum-normalized (over all documents in R) retrieval score of a document computed by a retrieval model, was used instead of $P(D\|Q)$, for the corresponding retrieval model.
WIG	The retrieval score of a document computed by a retrieval model, was used instead of $Score_{QL}(Q, D)$, for the corresponding retrieval model. The collection score, $Score_{QL}(Q, C)$, is always computed using the typical query likelihood model. We consider the *expanded queries* when computing this predictor using any of the query expansion models.
NQC	The collection score, $Score(Q, C)$ is always computed using the typical query likelihood model. We consider the *expanded queries* when computing this predictor using any of the query expansion models.
NSD & QTC-based	We consider the *expanded queries* when computing these predictors using any of the query expansion models.

Given this study setup, we ran extensive experiments that we discuss in details in the coming chapter.

Chapter 4

Experimental Evaluation

In this chapter, we first present our evaluation setup with details on the datasets and retrieval models used, and an overview on how evaluation of retrieval and prediction can be done. In Sections 4.2-4.7, we present and discuss in details, the results of evaluation of prediction in response to the six research questions studied in this thesis.

4.1 Setup

In this section, we discuss the evaluation setup we followed to evaluate our prediction framework. We first present the microblog datasets we used in this study. Next, we briefly discuss how retrieval effectiveness can be evaluated. We finally discuss the evaluation of prediction and how this intersects with retrieval evaluation.

4.1.1 Datasets

We conducted our experiments with two widely-used TREC tweets collections: **Tweets2011** [44] and **Tweets2013** [35]. Table 4.1 below briefly presents both test collections. Along with Tweets2011 collection, we used a merged set of the queries provided by TREC-2011 and TREC-2012 microblog tracks [44, 55]. As for Tweets2013, we used the queries provided by TREC-2013 microblog track [35].

Both collections are accessible remotely through a search API[1] provided by the microblog track organizers [35], who also made the collection statistics for both available.

The queries distributed in microblog tracks are short (3.10 and 3.28 words on average for Tweets2011 and Tweets2013 respectively). These queries resemble title-only queries in typical TREC collections typically used in ad-hoc search in related QPP studies [6, 63, 52, 9, 54].

Table 4.1: Tweets test collections used in our experiments.

Collection	Tweets	Time(days)	Queries	Source
Tweets2011	16M	16	108	TREC'11-12
Tweets2013	243M	59	60	TREC'13

[1] https://github.com/lintool/twitter-tools/wiki/TREC-2013-API-Specifications

Though these collections are composed of the same type of documents, i.e., tweets, yet the two collections are very different. In terms of size, Tweets2013 is around 15 times as large as Tweets11. Tweets2013 spans a much longer period in time which might indicate that it is more representative of the actual temporal distribution of tweets in real-life settings. Another important observation is that the average sampling rate in Tweets2013 is almost 4 times as large as the average rate in Tweets2011. This indicates that we are observing a different temporal distribution of tweets across the two collections which can also affect the temporal distribution of the relevant documents to queries across collections. Finally, the number of queries available with Tweets2011 is almost double of them with Tweets2013.

4.1.2 Retrieval

Given a document collection, a query Q submitted to a retrieval system, and a ranked list of results (documents) R retrieved for that query, retrieval *effectiveness* can be evaluated by estimating how relevant the documents are to the information need behind Q [51]. Working with a typical *test collection*, a query is usually associated with a set of *relevance judgments* that indicate which documents in the collection are relevant to the query. Given these judgments and the ranked list R, retrieval effectiveness can be evaluated using an *evaluation measure*. We focus on two measures that were the primary measures reported in the TREC'11-13 microblog tracks [44, 55, 35]. Mean Average Precision (MAP) is one such measure; it is computed as the mean of per query *average precision* (AP) values (usually computed at cut-off 1000) over all queries evaluated. Precision at rank 30 (P@30) is another retrieval evaluation measure.

Precision at rank k for a query can be computed as follows:

$$P@k(Q, R_k) = \frac{number\ of\ relevant\ documents\ \in R_k}{k} \tag{4.1}$$

where R_k is a sub-list of the result list R, with documents of rank$\leq k$.

The average precision is computed as follows:

$$AP(Q) = \frac{1}{|R_Q|} \sum_{D_r \in R_Q} P@k_{D_r} \tag{4.2}$$

where R_Q is the set of relevant documents to Q, D_r is a relevant document in R_Q ranked at rank k_{D_r} in the *original* result list R. To compute AP at cut-off 1000, we only consider relevant documents with $k_{D_r} \leq 1000$.

To study the performance of predictors across different retrieval models, we selected four effective retrieval models to consider with prediction (models are discussed in Section 3.4). In Table 4.2 we summarize the parameters used in each model in addition to the parameters values selected. We also highlight whether a model is a query expansion model, a temporal one, or both.

In Table 4.3, we report MAP and P@30 of the four models across both Tweest2011 and Tweets2013 collections.

Table 4.2: Summary on the retrieval models used with prediction.

Model	Temporal?	Expansion?	Parameters
QL	✘	✘	μ=2500
***t*-EXP**	✓	✘	r=0.01
QE	✘	✓	m=25, k=5
***t*-QRM**	✓	✓	m=5, k=25

Table 4.3: MAP and P@30 values for the retrieval models over Tweets2011 and Tweets2013. Measures of the model with best MAP over a collection are boldfaced. Second bests are surrounded by parentheses.

	Tweets2011		**Tweets2013**	
Model	MAP	P@30	MAP	P@30
QL	0.2957	0.3809	0.2677	0.4739
***t*-EXP**	(0.3026)	(0.3889)	0.2757	0.4789
QE	**0.3334**	**0.4247**	**0.3040**	(0.4967)
***t*-QRM**	0.2845	0.3478	(0.2979)	**0.5033**

4.1.3 Prediction

Usually, the effectiveness of prediction is evaluated using correlation; Pearson's r, Kendall's Tau τ, and Spearman's Rho ρ are among the most commonly-used correlation coefficients in QPP literature [3]. Using correlation, the performance of a predictor p is evaluated as follows. Given a set of queries, each is associated with (a) a *predicted* retrieval effectiveness value computed by a predictor and (b) an *actual* retrieval effectiveness value measured by a retrieval effectiveness measure (e.g., AP at cut-off 1000 discussed in section 4.1.2), correlation is computed using the sets of predicted and actual effectiveness values of all queries.

In this work, we use Pearson's r correlation coefficient to measure the quality of each predictor, considering both AP and P@30. Given a sample of n actual retrieval effectiveness values, $y = (y_1, ..., y_n)$, and the corresponding n predicted effectiveness values $\widehat{y} = (\widehat{y_1}, ..., \widehat{y_n})$, Pearson's correlation can be computed as follows:

$$r(y, \widehat{y}) = \frac{\sum_{i=1}^{n}(y_i - avg(y)) \cdot (\widehat{y_i} - avg(\widehat{y}))}{(n-1)\sigma_y \sigma_{\widehat{y}}} \tag{4.3}$$

where $avg()$ and σ are the sample mean and standard deviation, respectively.

4.1.4 Training and Testing

Earlier work on QPP has showed that prediction is dependednt on free parameters setting of predictors [6, 22, 54]; therefore, parameter tuning is needed to optimize prediction quality. To evaluate the quality of the predictors considering tuning of predictors' parameters, we adopted a train-test approach proposed by Shtok et al. [54]. We randomly split a query set into two subsets: a *training* (i.e., *tuning*) subset with 75% of queries and a *testing* subset with the remaining 25%. We tuned the free parameters of the predictors (by optimizing Pearson's coefficient) over the training subset, and then tested

the optimized predictors over the testing subset. With the relatively small number of queries per collection, using a single testing subset can result in a biased evaluation of prediction quality. To avoid bias, we repeated this (split-tune-test) process 120 times. The final prediction quality of a predictor is measured by averaging the correlation values (computed as discussed in Section 4.1.3) over the 120 splits.[2] Two-tailed paired t-test, with a significance level $\alpha = 0.05$, is used to determine statistically-significant differences in quality of the predictors [48, 56].

In this work, we compute the average of correlation values as follows:

$$Average\ r = \frac{\sum_{i=1}^{n} r_i}{n} \tag{4.4}$$

where r_i is the observed correlation for a split, and n is the number of splits. Average of Pearson's correlation values can be computed in other ways as well. One of the common ways of averaging correlations is based on back-converting the average of the Fisher's z [16] transformations of the correlation values observed [5]. We leave investigating the prediction quality based on that approach to future work.

While it is theoretically possible to follow a different evaluation approach in which we tune the predictors' parameters on one of the tweets collections and run the predictors on the other, we chose not to due to the large differences between the collections (refer to Section 4.1.1 for a comparison of the two collections).

In tuning the parameters for post-retrieval predictors (no parameters for pre-retrieval ones), we optimized correlation considering a wide range of values for each parameter. In Table 4.4, we report the parameters used for predictors and the ranges of values we optimized on.

Table 4.4: Parameters and ranges of values used in tuning.

Predictor	**Parameter**	**Description**	**Range (Step)**
All predictors	l	Length of result list	5-500 (5)
NSD	x	Stopping criterion to select list length	5-95 (5)
t-CLR	λ	Smoothing factor in $P(t\|Q)$	5-100 (5)
t-CLR, trm-CLR	h	Unit of time of timestamps (hours)	6-30 (6)
TTC-, IdfTTC-, LIdfTTC-based	m	Number of top terms	5-15 (5)

We briefly discussed how prediction can be evaluated given a test collection and a retrieval model. We finalize this section by listing the research questions that guided our work and then we present potential answers to these questions by presenting the evaluation results.

[2]We tried different splitting percentages and different number of splits, but found the reported setting to produce the best results.

4.1.5 Research Questions

We have the following 6 research questions in this study.

RQ1: How well do the state-of-the-art predictors perform in the context of microblog search? We consider predictors that are:

- Non-microblog-specific: predictors originally designed for retrieval contexts other than microblog
- Microblog-specific: predictors originally designed in the context of microblog search

RQ2: Can we improve QPP in the context of microblog search?

RQ3: Will the predictors' performance be consistent across different retrieval models that are used in microblog search? Specifically:

- Standard QL model
- Temporal models
- Query Expansion-based ones

RQ4: Will their performance be consistent across different test collections?

RQ5: Will predictors generalize to different retrieval performance measures?

RQ6: Can we improve prediction quality in the studied domain using a combination of predictors?

4.2 Evaluating Existing Predictors (RQ1)

In this section, we report and discuss evaluation of prediction quality of *existing* predictors. Results are Pearson's coefficient values based on Tweets2011 and considering both post- and pre-retrieval predictors. Unless otherwise specified, the results reported next are based on predicting the **average precision** (AP) at cut-off 1000 of retrieval results. Predictors are categorized into *families* of predictors. Due to the small number of pre-retrieval predictors, and their distinct prediction approach compared to post-retrieval ones, we group them in a separate family. We omit reporting results of MaxSCQ, AvgSCQ, and SCS due to their very weak correlation (less than 0.1150) over all retrieval models.

Table 4.5 presents full results on quality of all existing predictors with Tweets2011 and across all retrieval models.

Table 4.5: Pearson's correlation coefficient values for all predictors using Tweets2011. Best predictor per model is boldfaced.

	Non-microblog Post-retrieval				
	NQC	**WIG**	**NSD**	**CLR**	***t*-CLR**
QL	0.4028	0.3864	0.3889	**0.5197**	0.5000
***t*-EXP**	0.4053	0.4351	0.3632	0.4879	**0.5076**
QE	**0.5423**	0.4818	0.4922	0.1079	0.3613
***t*-QRM**	0.4914	-0.0633	**0.5130**	0.3241	0.4267

	Non-microblog Pre-retrieval					
	SumIdf	**MaxIdf**	**AvgIdf**	**VarIdf**	**DevIdf**	**SumSCQ**
QL	**0.3196**	0.2641	0.1365	0.2434	0.2844	0.2455
***t*-EXP**	0.3023	0.2641	0.1215	0.2720	**0.3123**	0.2352
QE	**0.3526**	0.3346	0.1975	0.2594	0.2882	0.2504
***t*-QRM**	**0.3561**	0.2818	0.1646	0.2178	0.2431	0.2694

	QTC-based						
	MeanQTC	**MedQTC**	**MinQTC**	**MaxQTC**	**DiffQTC**	**UpQTC**	**LowQTC**
QL	0.1019	0.0821	0.0844	0.0477	**0.2603**	0.0406	0.1205
***t*-EXP**	0.0967	0.0638	0.0992	0.0509	**0.2676**	0.0306	0.1084
QE	0.4207	0.3991	0.3487	0.3901	0.3923	0.3757	**0.4232**
***t*-QRM**	0.4862	0.3859	0.4645	0.3191	0.2872	0.3332	**0.5014**

	TTC-based						
	MeanTTC	**MedTTC**	**MinTTC**	**MaxTTC**	**DiffTTC**	**UpTTC**	**LowTTC**
QL	0.5150	**0.5458**	0.2613	0.3194	0.2630	0.4417	0.3852
***t*-EXP**	0.4921	0.4562	0.3568	0.3383	0.1555	0.3799	**0.4987**
QE	0.4002	0.3703	**0.4053**	0.2556	0.1463	0.2116	0.3286
***t*-QRM**	0.4960	0.5073	0.3742	0.3379	0.2382	0.4499	**0.5298**

	TCH-based						
	MeanTCH	**MedTCH**	**MinTCH**	**MaxTCH**	**DiffTCH**	**UpTCH**	**LowTCH**
QL	-0.1211	-0.0904	-0.0841	**-0.3363**	-0.3290	-0.0483	-0.1028
***t*-EXP**	-0.0653	-0.0431	-0.1032	-0.2228	**-0.2326**	-0.0955	-0.0333
QE	**-0.3127**	-0.2663	-0.0858	-0.2550	-0.2615	-0.2964	-0.2710
***t*-QRM**	-0.1227	-0.0353	0.0290	**-0.2705**	-0.2217	-0.0721	-0.0748

	Tweet-specific	
	#sRate	**URLsRate**
QL	0.0930	**0.3282**
***t*-EXP**	0.1534	**0.2600**
QE	0.1135	**0.2677**
***t*-QRM**	0.0779	**0.3927**

4.2.1 Non-microblog-specific Predictors

As noticed from Table 4.5, it is not possible to find one predictor that works best with all retrieval models.

- CLR was the best performing predictor with the QL model followed by t-CLR that had a much superior prediction quality compared to the remaining ones in this set (including both pre- and post-retrieval predictors).
- t-CLR outperformed all other predictors with the t-EXP model showing robustness in prediction across QL and t-EXP.

- NQC was the best performing predictor with the QE model and in fact it performed best over this model compared to its performance over all other models. NSD, that follows a similar intuition to NQC, came next outperforming all other predictors with this model. And NSD continued to exhibit good performance, being the best performing predictor over the t-QRM model.

In Table 4.5, we see that *idf*-based predictors are the best performing predictors compared to all pre-retrieval predictors tested. An interesting observation drawn from the table is that SumIdf outperformed CLR and WIG with the t-QRM model. It also outperformed CLR and had comparable performance to t-CLR with the QE model. This relatively strong performance of SumIdf compared to post-retrieval predictors is in line with findings of Shtok et al. [54] over ClueWeb09 Web collection.

We provide a rough comparison between the performance of these predictors in microblog search to their performance in other contexts. We focus our discussion on ad-hoc search using the QL model in the context of news and Web documents, and rely on Pearson's coefficient values reported by Shtok et al. [54] for CLR, NQC, WIG, and some pre-retrieval predictors.[3] We consider that study due to the large span of collections it covered (7 collections) in addition to the similar evaluation approach it followed.[4] In Figure 4.1, we plot the *range* of Pearson's correlation values for each of the studied predictors over all collections studied in [54]. The figure shows that the performance of the studied non-microblog-specific predictors in the context of microblog search generally lies in the range of their performance in the context of ad-hoc search over other collections.

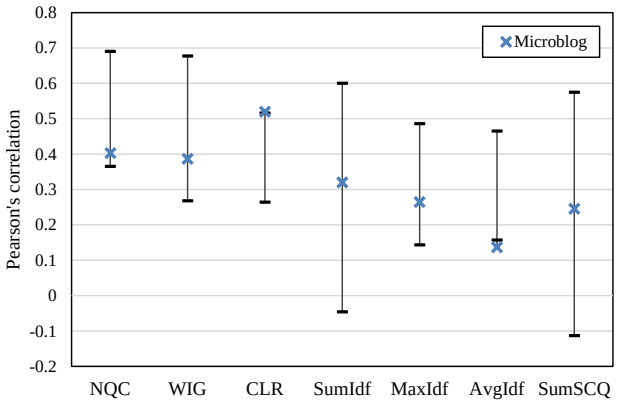


Figure 4.1: Pearson's correlation values for non-microblog-specific predictors in different contexts versus Microblog search.

4.2.2 Microblog-Specific Predictors

In this section, we study the predictors proposed by Perez and Jose [50] in the context of Microblog search. In Figure 4.2, we compare these with the non-microblog-specific predictors by comparing the overall best performing predictor per model for each. The

[3] No reference data is available for NSD and t-CLR.

[4] Keeping in mind different implementation details of the predictors, retrieval model, and different parameters setting in split-tune-test approach.

figure shows that, only with two models, QL and t-QRM, the microblog-specific predictors outperformed existing non-microblog-specific ones. With QL, MedTTC had a significantly higher performance compared to CLR with that model. With t-QRM, the difference was not significant comparing LowTTC with NSD. However, LowTTC had a lower performance compared to t-CLR with the t-EXP model. The performance of LowQTC was significantly lower than NQC with the QE model. This indicates that the microblog-specific predictors are not always the best to fit microblog search in their current design.

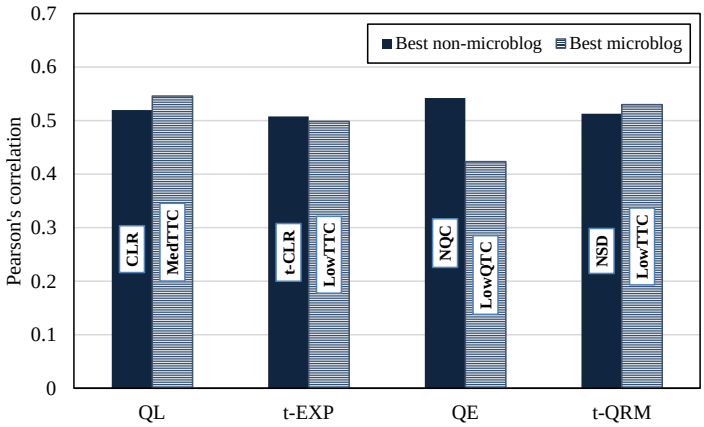


Figure 4.2: Pearson's correlation values for best non-microblog and microblog-specific predictors. Name of best predictor per model is on each bar.

An interesting observation from Figure 4.2 is that TTC-based predictors (median and lower TTC-based predictors specifically) were the best performing microblog-specific predictors with three out of four retrieval models. This conforms with the superior prediction quality of TTC-based predictors reported by Perez and Jose [50].

Surprisingly, TCH-based predictors were not performing as good as expected considering the temporality of tweets and microblog search. In Figure 4.3 we compare the best TCH-based predictor for each model, with the best predictor per model for each of the microblog-specific families. We report absolute values of Pearson's correlation since we are interested in comparing the *magnitude* of correlation across predictors.

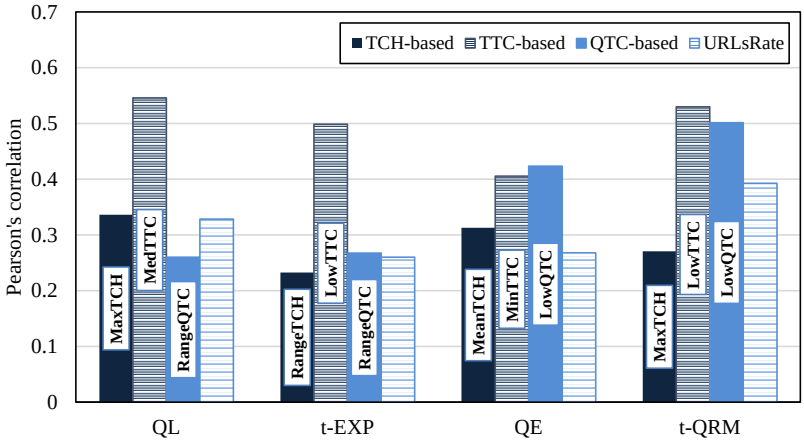


Figure 4.3: Pearson's correlation values for best microblog-specific predictors.

The figure shows that a TCH-based predictor outperformed two microblog-specific predictors only with the QL model, and outperformed one predictor with the QE model. Microblog-specific predictors that considered the tweet *content* generally had a superior performance over TCH-based predictors. Furthermore, comparing prediction quality of TCH-based predictors to the other temporal predictor we tested, i.e., t-CLR, we see that TCH-based predictors perform significantly worse. This might indicate that considering *surface* temporality of tweets relying on the tweet timestamp only in isolation from any other relevant information (temporal or non-temporal) might not be enough to produce good prediction in such temporal domain.

Another possible justification for the generally low performance of TCH-based prediction, is that TCH measures difference between tweets linearly to eventually capture temporal cohesion of the result list. This intuition assumes high cohesion indicates an easy-to-answer query since relevant tweets to the query will probably be found close to each other in time. While in fact, relevant tweets to a query will possibly cluster together in time, but might not be distributed linearly (nor uniformly) [14] in the time domain. Thus, this measure may be failing in accurately capturing the cohesion of the list.

We can also observe in Figure 4.3 that URLsRate (a tweet-specific predictor) is showing notable correlation with some retrieval models. It outperformed the best performing QTC-based predictor with the QL model, and had a relatively good correlation with the t-QRM model. It is also showing robustness across retrieval models. For such a simple predictor that basically tracks the appearance of URLs in the result list, it is showing some interesting results. The usefulness of considering URLs in prediction in microblog context is worth further investigation.

In Chapter 3, we discussed the intuition behind using the expanded queries in predictors that rely on query terms. In Figure 4.4 we compare the performance of some of the existing predictors that consider the query, considering expanded versus unexpanded queries. We mainly report results on the LowQTC predictor since it was the best performing microblog-specific predictor with the QE model. We also report comparison results on the NSD as one of the non-microblog-specific predictors that consider the query terms. We report absolute values of Pearson's correlation since we are interested in comparing the magnitude of correlation across predictors.

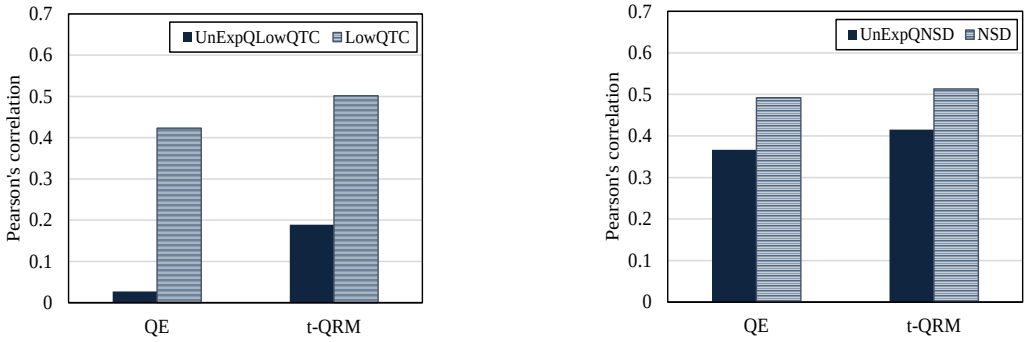


Figure 4.4: Pearson's correlation comparing predictors using expanded and unexpanded queries.

It is clear from Figure 4.4 that using the expanded queries highly improved prediction with these predictors. In fact, using expanded queries helped *significantly* improve prediction with the expansion models for both predictors. We also observed such improvement

considering other predictors working with the query terms such as NQC. Even with a predictor that considered the query by using the query length only (like the case with NSD), using the expanded query significantly improved prediction. This indicates that considering the expanded queries in prediction given results retrieved through a query expansion model, can provide additional information to help improve the prediction approach.

We have briefly discussed the performance of the existing microblog-specific and non-microblog-specific predictors over the four retrieval models we worked with. Results have shown that microblog-specific predictors are not significanlty improving prediction in the current context compared to existing non-microblog-specific predictors. We also provided results that encourages using expanded queries in prediction with query expansion retrieval models.

4.3 Evaluating Proposed Variants (RQ2)

We focused our discussion in the previous section on evaluating the performance of existing predictors. In this section, we discuss the evaluation of our proposed variants of some of the existing predictors. We mainly discuss the evaluation of the two *idf*-based QTC variants, the two *idf*-based TTC variants, ExpTCH-based predictors, and finally *trm*-CLR.

TTC- and QTC-based predictors were the best performing microblog-specific predictors. In Figure 4.5, we compare the performance of these predictors with the performance of the two corresponding *idf*-based variants of each. For example, we compare the MedTTC (best with QL model) with MedIdfTTC and MedLIdfTTC.

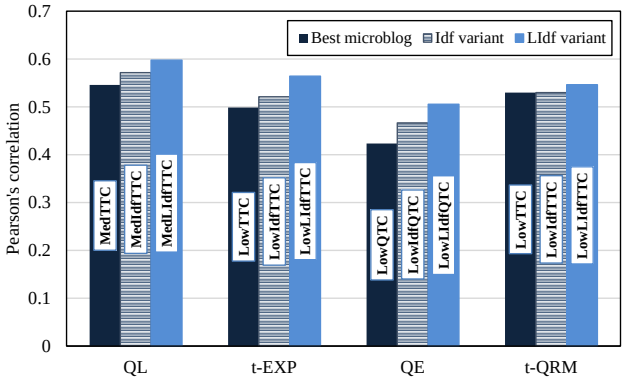


Figure 4.5: Pearson's correlation values for best microblog-specific predictors and their *idf*-based variants.

The figure clearly shows that for each model, both variants outperformed the corresponding existing microblog-specific predictor. Only in one occasion, the variant LowIdfTTC was almost indifferent from LowTTC with the *t*-QRM model. We can also observe that using query length normalization with the *idf*-based TTC and QTC (LIdfTTC, and LIdfQTC) measure helped achieve the best prediction quality among the three variants. To get a sense of the actual difference in numbers, we present the Pearson's correlation values in Table 4.6. The numbers demonstrate the significance

of the proposed variants in improving prediction quality of TTC- and QTC-based predictors.

Table 4.6: Pearson's correlation coefficient values for proposed variants of best microblog-specific predictors. Best predictor per model is boldfaced. Value marked with ** indicates a highly significant improvement over original corresponding predictor, $p < 0.01$.

Model	Best Predictor Name	Best Predictor	Idf Variant	LIdf Variant
QL	MedTTC	0.5458	0.5718**	**0.5976****
***t*-EXP**	LowTTC	0.4987	0.5209**	**0.5642****
QE	LowQTC	0.4232	0.4665**	**0.5056****
***t*-QRM**	LowTTC	0.5298	0.5297	**0.5463**

In Figure 4.6, we compare the performance of the best proposed variant from Table 4.6 to performance of the best performing, non-microblog predictors over all retrieval models. We notice that the LIdf-based variant is now outperforming non-microblog predictors over all models except for QE. The improvement was significant with QL and t-EXP.

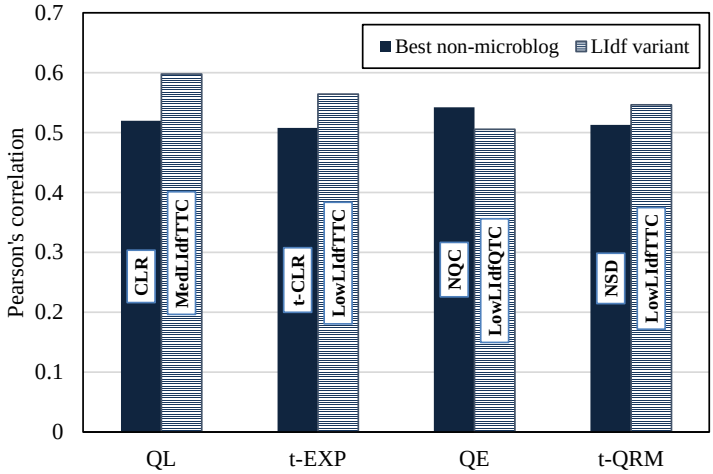


Figure 4.6: Pearson's correlation values for best non-microblog-specific predictors and LIdf-based variants.

We now shed some light on the overall performance of LIdf-based predictors, considering both TTC- and QTC-based ones. LIdf-based predictors which are variants corresponding to the original best performing TTC- and QTC-based predictors managed to outperform original ones. However, our results show that some other LIdf-based predictors were actually the best performing among all LIdf-based predictors. In Table 4.7, we report the best performing LIdfQTC- and LIdfTTC-based predictors and compare them to the original, best performing QTC- and TTC-based ones. We also compare them to best performing non-microblog ones.

Table 4.7: Pearson's correlation coefficient values for best performing LIdfQTC- and LIdfTTC-based predictors. Best LIdf-based predictor outperforming best non-microblog and original microblog is boldfaced. Value marked with a and/or b indicates a significant improvement over original corresponding predictor and/or best non-microblog, respectively, $p < 0.05$.

Model	Non-microblog	QTC-based	TTC-based	LIdfQTC-based	LIdfTTC-based
QL	0.5197 (CLR)	0.2603	0.5458	0.3722^{a} (Range)	$\mathbf{0.5976}^{a,b}$ (Median)
***t*-EXP**	0.5076 (t-CLR)	0.2676	0.4987	0.3728^{a} (Range)	$\mathbf{0.5642}^{a,b}$ (Lower)
QE	0.5423 (NQC)	0.4232	0.4053	0.5114^{a} (Mean)	0.4868^{a} (Minimum)
***t*-QRM**	0.5130 (NSD)	0.5014	0.5298	$\mathbf{0.5550}^{a,b}$ (Mean)	$\mathbf{0.5655}^{a,b}$ (Median)

Table 4.7 shows that the proposed LIdfTTC-based predictors are relatively strong predictors, with the best of them significantly outperforming best original TTC-based and non-microblog ones with all models but QE.

We now compare the performance of the best predictor of our ExpTCH variant to the best TCH-based predictor for each model, as summarized in Table 4.8.

Table 4.8: Pearson's correlation coefficient values for best performing TCH- and ExpTCH-based predictors. ExpTCH-based predictor outperforming TCH-based is boldfaced. Value marked with * indicates a significant improvement over TCH-based predictor, $p < 0.05$.

Model	TCH-based	ExpTCH-based
QL	-0.3363	**0.3659**
***t*-EXP**	-0.2326	**0.2630**
QE	-0.3127	**0.3174**
***t*-QRM**	-0.2705	$\mathbf{0.3987}^{*}$

The table shows that the best ExpTCH-based predictor outperforms the best TCH-based one with all models. The improvement was significant with the t-QRM model. Overall, we see that our proposed ExpTCH-based predictors are positively correlated with average precision while the original TCH-based ones are negatively correlated. This is justifiable since the TCH measure is correlated to the actual difference between tweets' timestamps. The higher the time difference, TCH values are higher. ExpTCH is lower in this case, since it is computed as a measure to *reward* small time difference between two documents by producing a high ExpTCH value. Considering the improvement of prediction quality with our proposed ExpTCH-based predictor, we think that this predictor is probably better to fit prediction in the context of microblog search.

We briefly discuss the performance of our last proposed variant, i.e., *trm*-CLR. We mainly focus on comparing its performance to the performance of the CLR predictor as shown in Figure 4.7. The proposed predictor performed relatively poorly especially with the QE model. Further invistgation is needed to justify such bad performance of this predictor.

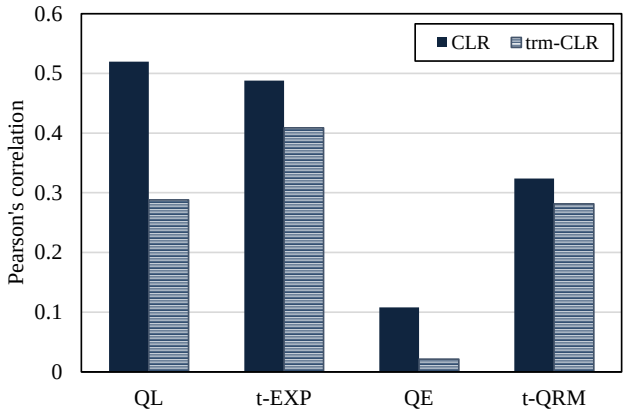


Figure 4.7: Pearson's correlation values for *trm*-CLR and CLR.

4.4 Evaluating Predictors across Retrieval Models (RQ3)

Our goal in this section is to study the consistency of predictors across retrieval models. We first start by comparing performance of existing non-microblog predictors across retrieval models as Table 4.9 shows.

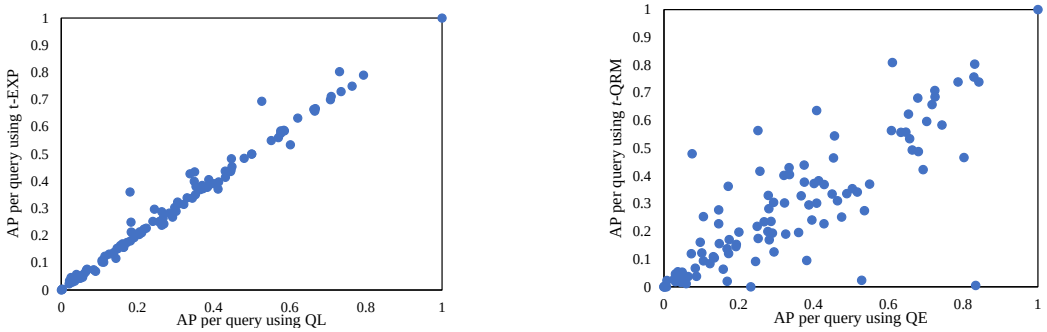
Table 4.9: Pearson's correlation coefficient values for all non-microblog post-retrieval predictors. Best predictor per model is boldfaced.

Model	NQC	WIG	NSD	CLR	*t*-CLR
QL	0.4028	0.3864	0.3889	**0.5197**	0.5000
***t*-EXP**	0.4053	0.4351	0.3632	0.4879	**0.5076**
QE	**0.5423**	0.4818	0.4922	0.1079	0.3613
***t*-QRM**	0.4914	-0.0633	**0.5130**	0.3241	0.4267

We notice that NQC and NSD performed significantly better with expansion models compared to their performance with the non-query expansion models, i.e., QL and *t*-EXP. In contrary to that, CLR and *t*-CLR performed well with the non-query expansion models. One possible reason to this drop in performance (especially with CLR), is possibly the noise introduced to the results of some queries that do not benefit from expansion. It might be the case that query expansion degraded the ability of these two predictors to capture the true performance of a query by the reduction of *coherence* in the list due to introducing new documents and possibly noise.

In this family, *t*-CLR was the only temporal predictor tested. We compare the performance of this predictor across models considering two groups of models: query expansion and non-expansion based models (refer to Table 4.2 for details on the models). In non-query expansion models, we notice that *t*-CLR had a better quality with the temporal model *t*-EXP compared to the non-temporal model QL. As for expansion models, *t*-CLR performed significantly better with the temporal model *t*-QRM compared to the non-temporal one, i.e., QE. This might be an indication that a temporal predictor better fits a temporal retrieval model.

Figure 4.8 helps justify why t-CLR did not have a significantly better performance with t-EXP compared to QL, yet the improvement was big with t-QRM compared to QL. In Figure 4.8, we correlate the retrieval effectiveness (measured by MAP) of the two models discussed, over all queries used with Tweets2011. It is obvious from Figure 4.8a that QL and t-EXP are highly correlated although t-EXP performs significantly better than QL (using two-tailed paired t-test, $\alpha = 0.05$). This might indicate that t-CLR was able to capture temporality of the data and the model but not by much due to the high correlation between t-EXP and QL initially. Figure 4.8b shows the lower correlation between the two models which might resulted in the significantly different (and higher in this case) prediction quality of t-CLR with t-QRM compared to QE.



a. Correlating QL and t-EXP. b. Correlating QE and t-QRM.

Figure 4.8: Correlation between retrieval models categorized by whether they are query expansion models or not.

As for microblog-specific predictors, we present some main results on examples of them in Table 4.10.

Table 4.10: Pearson's correlation coefficient values for best performing QTC- and LIdfQTC-based predictors.

Model	QTC-based	LIdfQTC-based
QL	0.2603	0.3722
***t*-EXP**	0.2676	0.3728
QE	0.4232	0.5114
***t*-QRM**	0.5014	0.5550

Table 4.10 indicates that QTC- and LIdfQTC-based predictors perform much better with query expansion models (QE, and t-QRM) compared to their performance with non-expansion ones. This can be reasonably justified by the fact that in these query expansion models, we are adding more terms to the query. These additional terms helped query-coverage-based predictors to better capture the topic in the resulting documents, and how related they are to the query. In fact, QTC-based predictors performed worse than best performing pre-retrieval predictors, and LIdf-based predictors were not much better with QL and t-EXP models. This indicates that query-coverage-based predictors will probably work better with query expansion models, and more generally longer queries, like the description associated with title queries in typical TREC collections.

4.5 Evaluating Prediction over other Test Collections (RQ4)

So far, we have discussed the performance of predictors in predicting the average precision of retrieval for queries over Tweets2011. In this section, we briefly present and discuss results of evaluating prediction over Tweets2013. We mainly focus on results related to the highlights and conclusions resulting from evaluating prediction over Tweets2011.

4.5.1 Non-microblog-specific Predictors

In Table 4.11, we present Pearson's coefficient values for existing non-microblog post-retrieval predictors. It should be noted that the prediction quality is generally lower for these predictors compared to their performance over Tweets2011. This might be due to the smaller number of queries used in evaluation over Tweets2013 which might hindered parameter tuning or testing of predictors. We skip discussing pre-retrieval predictors since there performance was generally poor.

Table 4.11: Pearson's correlation coefficient values for all non-microblog post-retrieval predictors over Tweets2013. Best predictor per model is boldfaced.

Model	NQC	WIG	NSD	CLR	*t*-CLR
QL	0.2412	0.3293	0.3925	**0.3941**	0.3025
***t*-EXP**	0.2490	0.3320	0.3567	**0.3811**	0.3379
QE	**0.5199**	0.3805	0.4083	0.1058	0.1460
***t*-QRM**	**0.4264**	-0.0789	0.3381	0.1656	0.3932

CLR was the best performing predictor over the two non-expansion models, QL and t-EXP. While NQC was the best performing one over expansion models. Almost in all predictors, we notice that the performance of each predictor is very similar with both QL and t-EXP. This can be a result of the high correlation between the two retrieval models as shown in Figure 4.8. NQC was also the best performing non-microblog-specific predictor over QE model with Tweets2011. Furthermore, its performance with Tweets2013 was not significantly different from its prediction quality with Tweets2011. This indicates that NQC is a relatively strong and robust predictor across the collections with QE model.

As pointed out earlier, we see that CLR is behaving relatively worse with expansion models compared to non-expansion ones. This conforms with a similar observation and discussion reported with Tweets2011 in Section 4.4. As for NSD, we notice that its performance is very robust across the two collections with the QL and t-EXP models specifically.

We also observe that the performance of the only temporal predictor here t-CLR is following a similar trend as discussed in Section 4.4 over Tweets2011. t-CLR performed significantly better with the temporal non-expansion model compared to the non-temporal one. The improved performance was also significant with the temporal query expansion model compared to the non-temporal one. Indeed, this is another indication that temporal predictors might better fit temporal models.

Interestingly, and differently from the case in Tweets2011, we notice that t-CLR is performing much better with the temporal expansion model t-QRM compared to QE and to the two non-temporal models. Moreover, its performance with t-EXP that is not very different from QL, is better compared to its performance with the latter. This result might again indicate that a temporal predictor better fits temporal models. Though no such observation on t-CLR holds with the Tweets2011 collection, we believe it is worth further investigation because of the different (and possibly more prevalent) temporal nature of Tweets2013 compared to Tweets2011 (refer to Section 4.1.1 for details).

In both Tweets2011 and Tweets2013, we notice that the performance of WIG severely drops with t-QRM model. It is less likely that this is happening because t-QRM is an expansion model since it had a relatively stable performance with QE. Yet, it is possible that because we modeled the collection using the QL model in WIG (Eq. 3.11) and used a completely different (not to mention temporal) model to model the query. This inconsistency in models and the heavy involvement of the collection score in computing WIG might resulted in this large drop.

4.5.2 Microblog-specific Predictors

In Figure 4.9, we report the best performing predictor per model categorized by family of predictor. We plot the absolute correlation of TCH-based predictors as we are only concerned with comparing the magnitude of prediction.

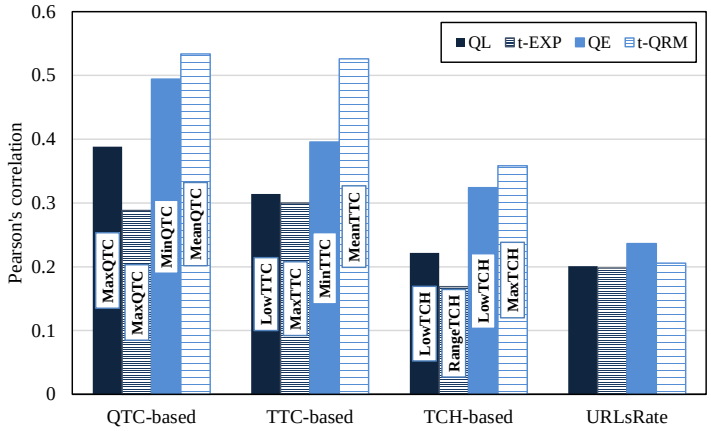


Figure 4.9: Pearson's correlation values for microblog-specific predictors over Tweets2013.

The figure shows that QTC-based predictors are generally outperforming all other predictors with almost all models. This is opposing the case with Tweets2011, where TTC-based predictors were the superior predictors. As observed with Tweets2011, URLsRate was the best performing tweet-specific predictor. Figure 4.9 also conforms the finding over Tweets2011 that this predictor is very robust across retrieval models.

The figure also shows that QTC-based predictors are performing well and much better with query expansion-based models compared to their performance with non-expansion ones. It is also interesting to observe that QTC-base predictors are performing better (and significantly better with some models) with Tweets2013 compared to Tweets2011 as shown in figure 4.10.

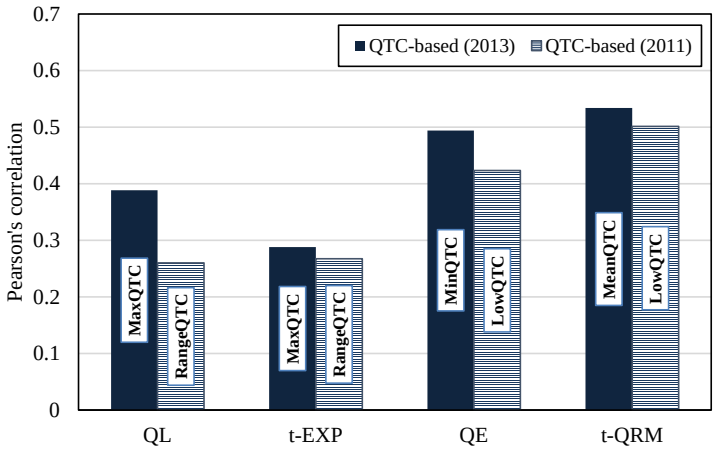


Figure 4.10: Pearson's correlation values for best QTC-based predictor over both Tweets2013 and Tweets2011.

Possibly, this difference in performance of QTC-based predictors across collections is due to large difference in the number of queries used with the two collections. Furthermore, we do not have an accurate comparison of the distribution of query categories across collections which might affect prediction across collections. Indeed, this motivates us to consider studying the performance of predictors per category of queries considering different categorization schemes such as temporality of queries, or aim of them.

In Figure 4.11, we plot the performance of microblog-specific predictors compared to non-microblog ones. The figure shows that only with one model, microblog-specific predictors outperformed non-microblog ones. Yet, the difference between these two types of predictors was not significant with two of the remaining models. These results in addition to the results of such comparison over Tweets2011, indicate that predictors proposed in the context of microblog search are not always the best to fit this context in their current design.

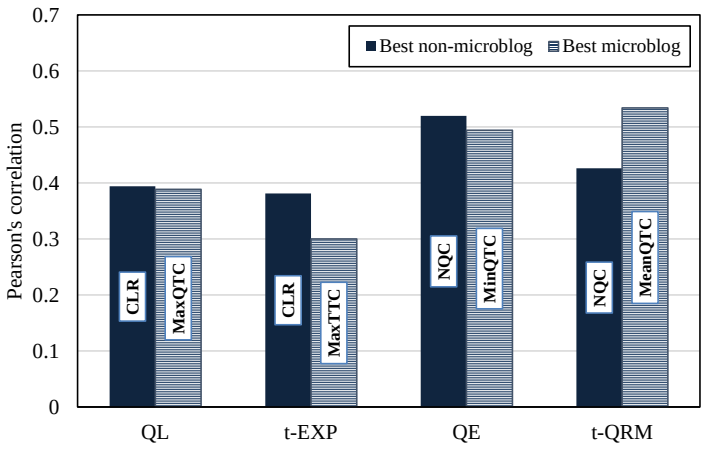


Figure 4.11: Pearson's correlation values for best non-microblog and microblog-specific predictors over Tweets2013. Name of best predictor per model is on each bar.

4.5.3 Evaluating Proposed Variants

We now focus on discussing the performance of the proposed variants to existing microblog- and non-microblog-specific predictors. We present the results of this comparison considering the Idf- and LIdf-based variants for both QTC- and TTC-based predictors in Table 4.12. We mainly focus on the best variant of QTC/TTC.

Table 4.12: Pearson's correlation coefficient values for best performing QTC-based variant and TTC-based variant with Tweets2013. Best variant outperforming best non-microblog and original microblog is boldfaced. Value marked with a and/or b indicates a significant improvement over original corresponding predictor and/or best non-microblog, respectively, $p < 0.05$.

Model	Non-microblog	QTC-based	TTC-based	QTC Variant	Name	TTC Variant	Name
QL	0.3941 (CLR)	0.3883	0.3143	0.3885	MaxIdfQTC	**0.4193**^{a}	MinLIdfTTC
***t*-EXP**	0.3811 (CLR)	0.2882	0.3000	0.2928	MaxIdfQTC	**0.4450**^{a,b}	MinLIdfTTC
QE	0.5199 (NQC)	0.4941	0.3954	0.5120	UpIdfQTC	0.4358^{a}	LowIdfTTC
***t*-QRM**	0.4264 (NQC)	0.5338	0.5261	**0.5370**^{b}	MeanLIdfQTC	**0.5744**^{a,b}	MinIdfTTC

The table shows that our TTC variants managed to significantly improve prediction compared to the best performing TTC-based predictor. They outperformed non-microblog predictor with three models, but the improvement was significant with two models only. The performance of TTC- and variant of TTC-based predictors performed significantly better with the t-QRM model.

As for variants of QTC, we observe a slight improvement over the original QTC-based predictor. Similar to QTC-based predictors, predictors based on the variant of QTC are performing significantly better with query expansion models compared to their performance with non-expansion ones.

In Figure 4.12, we compare the performance of the best performing TCH-based predictor to the best performing ExpTCH-based one with each model. Note that we plot the magnitude of correlation coefficient values to perform such comparison.

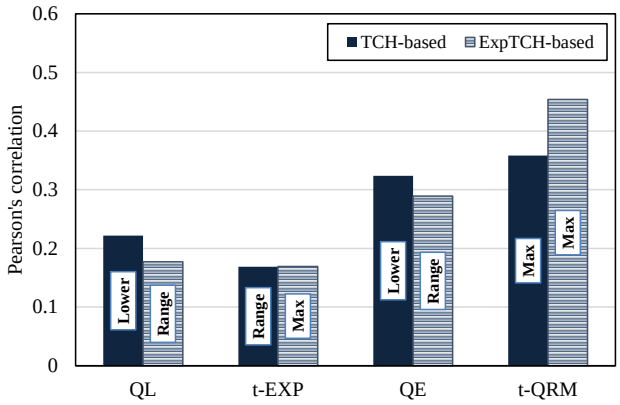


Figure 4.12: Pearson's correlation values for best TCH- and ExpTCH-based predictors over Tweets2013.

Differently from Tweets2011, we see that our proposed variant improved prediction over the TCH-based one with the t-QRM model only. In fact, and similar to the case with Tweets2011, the improvement with this model was significant. This might indicate that the ExpTCH-based predictors better captured the temporal distribution of results with this model which agrees with our intuition behind selecting this predictor.

As for our last proposed variant *trm*-CLR, it is performing relatively poorly with all retrieval models. Similar to Tweets2011, its performance was significantly worse than that of the CLR predictor as can be seen in Figure 4.13. The performance was particularly poor with the QE model.

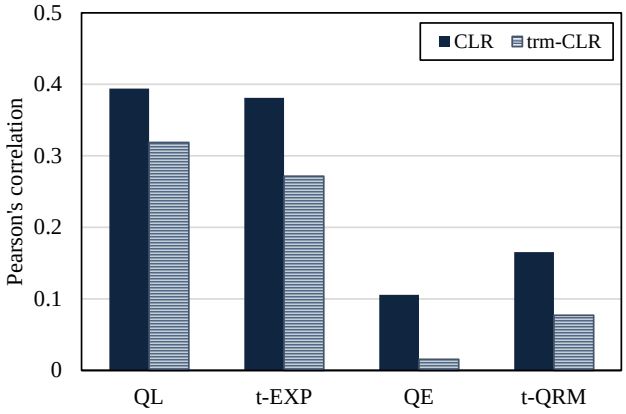


Figure 4.13: Pearson's correlation values for *trm*-CLR and CLR over Tweets2013.

4.6 Evaluating Prediction of other Retrieval Performance Measures (RQ5)

So far, we have been discussing the evaluation of prediction of the average precision (AP) as the retrieval performance measure. Precision at rank 30 (P@30) is another widely-used retrieval effectiveness measure in the context of microblog search. Therefore, we are interested in evaluating the performance of prediction of this measure.

Due to the larger number of queries with Tweets2011, we believe it is generally more representative of the prediction quality. Thus, we present discussion and results considering this test collection only.

4.6.1 Non-microblog-specific Predictors

Pre-retrieval predictors performed poorly in prediction of P@30 (with a Pearson's correlation coefficient that is generally less than 0.13), therefore we focus on discussing the performance of post-retrieval ones. Table 4.13 presents the prediction quality of the post-retrieval, non-microblog predictors.

Table 4.13: Pearson's correlation coefficient values for non-microblog post-retrieval predictors of P@30. Best predictor per model is boldfaced.

Model	NQC	WIG	NSD	CLR	*t*-CLR
QL	0.3467	0.3084	**0.3832**	0.2883	0.2490
***t*-EXP**	**0.3788**	0.3563	0.2968	0.2786	0.1814
QE	**0.4307**	0.4281	0.1835	0.0042	0.2899
***t*-QRM**	**0.3907**	-0.0444	0.2018	0.1183	0.2950

Some of the observations drawn from this table are listed below.

- NQC was the best performing predictor with all models but QL.
- NSD was the best performing predictor with QL followed by NQC.
- WIG had a severe drop in prediction performance with the t-QRM model.
- CLR had a large drop in performance with expansion models compared to non-expansion ones. On the contrary, t-CLR had better performance with expansion models.
- Performance of both NQC and WIG was significantly better than other predictors with the both t-EXP and QE models.
- NQC's performance was generally consistent across retrieval models.

We now compare the performance of the best and worst predictors (per model) in predicting P@30 to those in predicting AP in Figure 4.14. We plot the magnitude of correlation for each predictor, with the predictor's name on the bar reflecting its quality.

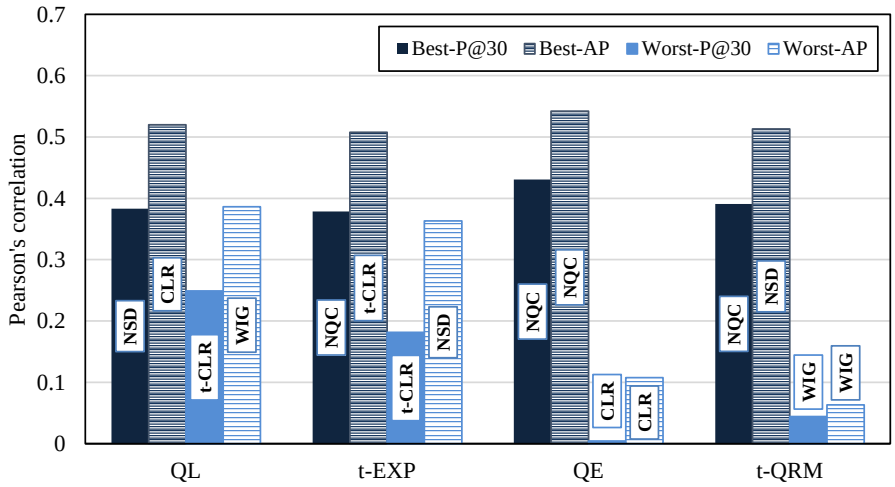


Figure 4.14: Pearson's correlation values for best and worst predictors in predicting AP and P@30.

The figure shows that CLR was the worst performing predictor in predicting both AP and P@30 with the QE model, and WIG was the worst in predicting both with t-QRM. The figure also shows that documents scores standard deviation-based predictors (namely, NSD and NQC) were the best in predicting both AP and P@30 with QE and t-QRM models. Specifically, NQC was the best in predicting AP and P@30 with the QE model, and NSD and NQC were the best in predicting AP and P@30, respectively, with t-QRM. However, when looking back to Table 4.5, we see that the difference in prediction quality of NSD and NQC in predicting AP with t-QRM was not significant. These observations indicate that the relative performance of some predictors can be consistent across performance measures with some retrieval models.

Another observation drawn from Figure 4.14 is that, the best achievable prediction quality in predicting P@30 was generally lower than that in predicting AP with all

models. However, the best prediction quality of P@30 over different models is relatively good and considerable with some models. Looking at this observation and observations made earlier indicates, that although the tested predictors were generally used to predict AP [3], they can predict other performance measures with a relatively good prediction quality.

4.6.2 Microblog-specific Predictors

In Figure 4.15, we present the best prediction performance in predicting P@30 for each of the families of microblog-specific predictors.

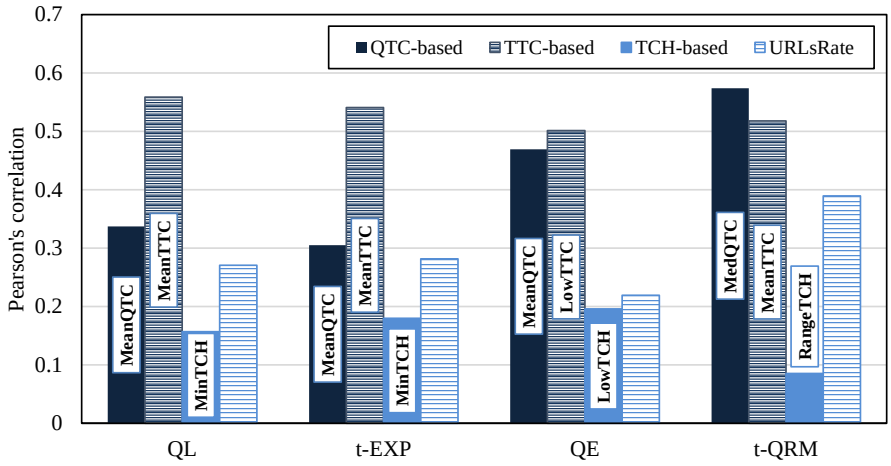


Figure 4.15: Pearson's correlation values for best predictor in predicting P@30 per microblog-specific family.

It is clear from the figure that we can predict P@30 with a good prediction quality across all retrieval models using microblog-specific predictors. More observations on this figure are listed below:

- A TTC-based predictor was generally the best in predicting P@30.
- Prediction quality of the best TTC-predictor was generally consistent across models.
- Best performing TCH-based predictors were the worst performing predictors across all models.
- Surprisingly, the simple URLsRate predictor had a considerable prediction quality with t-QRM.
- Similarly to the case with predicting AP (see Section 4.4), QTC-based predictors had a significantly better prediction quality in predicting P@30 with expansion models compared to non-expansion ones.
- Predictors that considered the actual tweet content were significantly better than TCH- and Tweet-specific-based predictors.

After this general discussion of results, we now compare the performance of microblog-specific predictors to non-microblog-specific ones in predicting P@30. We perform such comparison since we are interested in finding the best performing predictors across *all* families of existing predictors. Results are presented in Figure 4.16.

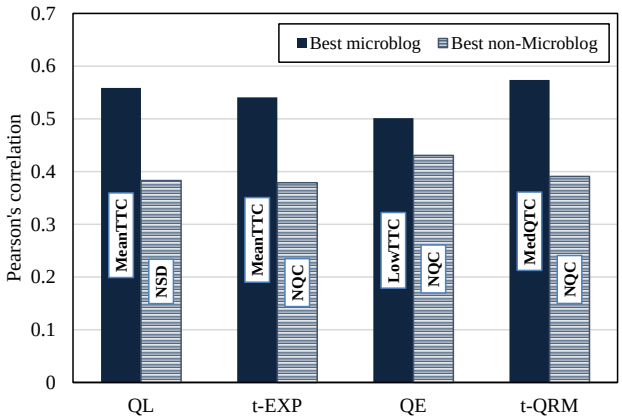


Figure 4.16: Pearson's correlation values for best microblog- and non-microblog-specific predictors in predicting P@30.

This figure shows that the best microblog-specific predictor per model was significantly better in predicting P@30 compared to the best non-microblog-specific one. Looking back to Figure 4.2, we see that this did not hold when predicting AP. In fact, when predicting AP, we could not conclusively decide that one group outperformed the other with all models. Further investigation is needed to explain this observation.

In Figure 4.17, we compare the best prediction performance in predicting P@30 to that of predicting AP. As the figure shows, best performing microblog-specific predictors

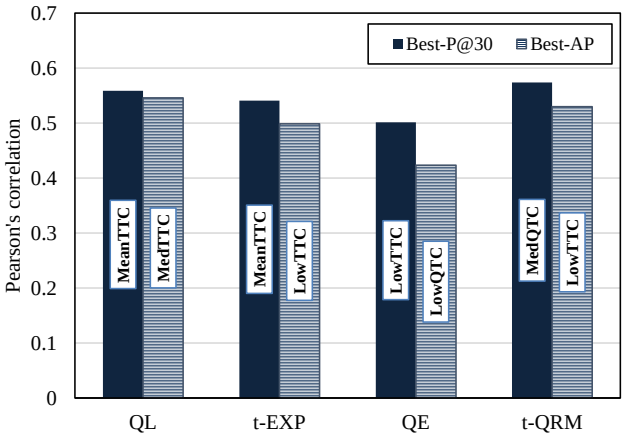


Figure 4.17: Pearson's correlation values for best microblog-specific predictors in predicting P@30 and AP.

managed to predict P@30 with a quality that is notably higher than its quality in predicting AP. The difference was significant with all models but QL. These results along those reported for prediction of P@30 using non-microblog-specific predictors are promising. The results again show that we can predict P@30 with a relatively good quality using existing predictors.

4.6.3 Evaluating Proposed Variants

In this section, we discuss the evaluation of our proposed variants in predicting P@30.

IdfQTC and LIdfQTC Variants

We first discuss the performance of both IdfQTC- and LIdfQTC-based variants. We mainly focus on comparing the performance of the *best* variant to the performance of the best QTC-based prediction. Such comparison is illustrated in Figure 4.18.

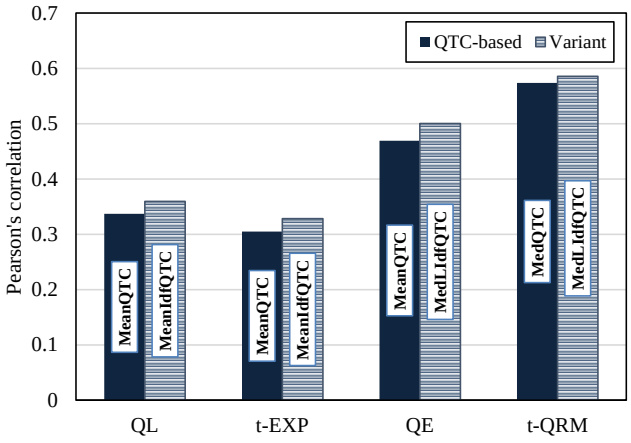


Figure 4.18: Pearson's correlation values for best predictors based on variants of QTC and best QTC-based predictors in predicting P@30.

The figure shows that MedLIdfQTC was the best performing predictor over expansion models and MeanIdfQTC was the best over non-expansion ones. We see that the proposed variants helped improve prediction quality over original, QTC-based predictors. However, the improvement was not significant with any of the models.

IdfTTC and LIdfTTC Variants

We now evaluate the predictors based on variants of TTC. In Figure 4.19, we plot the best performing TTC variant-based predictors compared to the best performing TTC-based ones. Similarly to the case with QTC variants, the variants of TTC helped improve prediction quality but only significantly with t-QRM. Moreover, we observe that the proposed variant resulted in a degraded prediction quality with the QL model.

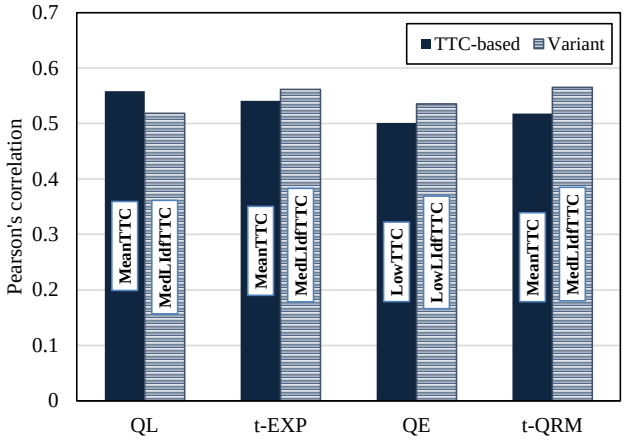


Figure 4.19: Pearson's correlation values for best predictors based on variants of TTC and best TTC-based predictors in predicting P@30.

ExpTCH Variant

Although TCH-based predictors did not have good prediction quality in predicting P@30 (see Figure 4.15), we are still interested in evaluating the improvement resulting from using our proposed variant, i.e., ExpTCH. Figure 4.20, compares the performance of the best ExpTCH-based predictor to the best TCH-based one per model.

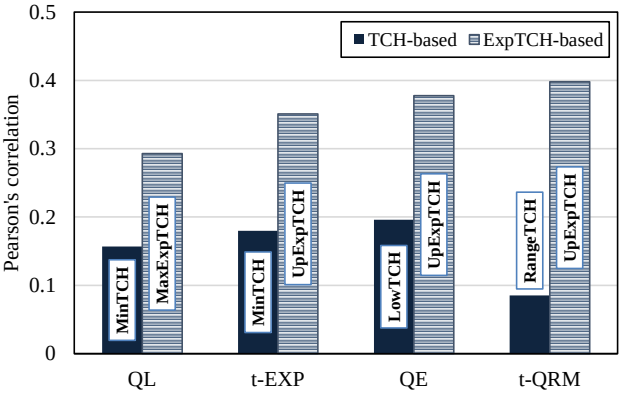


Figure 4.20: Pearson's correlation values for best predictors based ExpTCH and best TCH-based predictors in predicting P@30.

Surprisingly, we notice that our proposed variant allowed for significant improvement in quality of predicting P@30. It is interesting to see that, with the t-QRM model, the quality of the best performing TCH-based predictor was much lower than that of the ExpTCH-based one. This might indicate that the ExpTCH-based predictors managed to better capture the temporal distribution of the top 30 documents. Such observation still holds with other models as well.

The figure also shows that the performance of the best ExpTCH-based predictors was better with the temporal non-expansion model, QL compared to the temporal one, i.e., t-EXP. The same relative quality holds comparing the predictor's performance with temporal, expansion model to the non-temporal one. This might indicate that this temporal predictor better fits temporal models.

We are also interested in comparing the performance of the best ExpTCH-based predictors in predicting P@30 to the best prediction quality in predicting AP. Figure 4.21 provides such comparison.

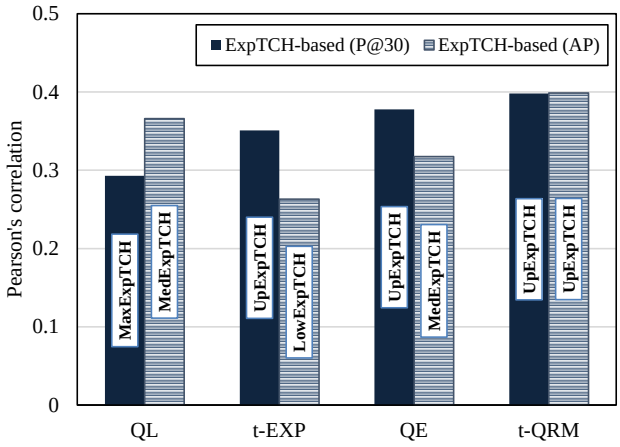


Figure 4.21: Pearson's correlation values for best predictors ExpTCH-based in predicting P@30 and AP.

As can be seen in Figure 4.21, the best performing ExpTCH-based predictors had a significantly better prediction quality in predicting P@30 with t-EXP and QE models compared to prediction quality of best performing ExpTCH-based predictions of AP. The difference was negligible with the t-QRM. With QL only, the ExpTCH-based predictor had better quality in predicting AP. These observations indicate that possibly, ExpTCH-based predictors can better predict P@30.

Due to the low prediction quality of *trm*-CLR (i.e., temporal variant of CLR) with almost all models, we skip discussing how it performed in predicting P@30.

We compare the best performing existing predictors to the performance of best performing variant-based ones in predicting P@30, as shown in Figure 4.22. We see that our proposed variants managed to improve prediction quality with all models but QL. Although improvement is not big, yet we see that such improvement is considerable.

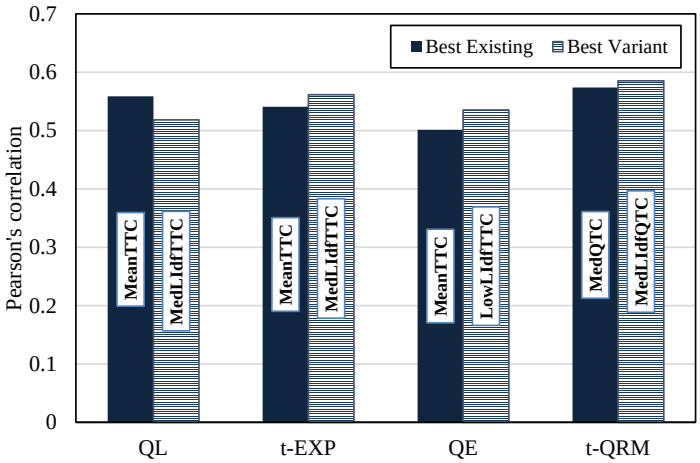


Figure 4.22: Pearson's correlation values for best existing and variant-based predictors in predicting P@30.

To wrap up the discussion on evaluation in response to research questions 1-5, we finally compare the best prediction quality in predicting AP to that in predicting P@30 across different retrieval models, over Tweets2011.

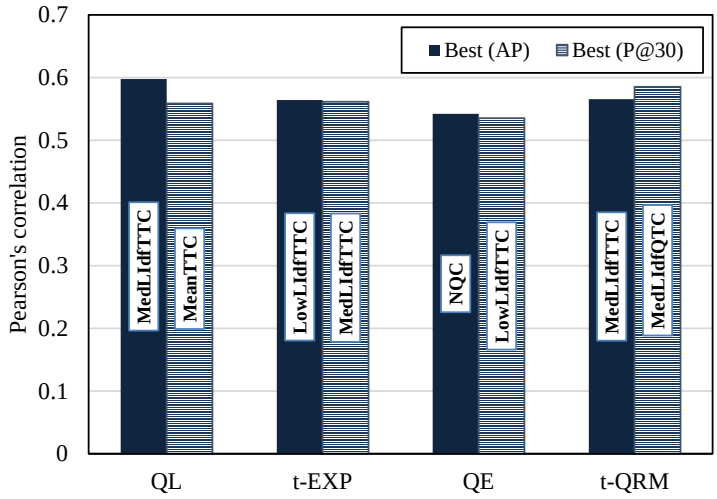


Figure 4.23: Pearson's correlation values for best predictors in predicting AP and P@30 over Tweets2011.

The figure emphasizes the following:

- Predicting both AP and P@30, with a relatively good prediction quality in a robust manner across different retrieval models, is possible in the context of microblog search.
- Predictors based on our proposed variants were generally the best in predicting both AP and P@30 across models.

4.7 Combining Predictors (RQ6)

Combining predictors showed notable improvements in prediction quality in previous studies [60, 10, 25, 63, 20, 53, 18, 50]. To improve prediction quality in this context, we also attempt this approach using linear regression to combine predictors. We used Weka's [17] implementation of linear regression in this task.

4.7.1 Experimental Setup

In our experiments, we consider all existing pre- and post-retrieval predictors, in addition to all predictors based on our proposed variants evaluated in our earlier experiments. Prediction and combination of predictors are performed with all four retrieval models, considering the AP only as the effectiveness measure to predict. Pearson's correlation coefficient is used as the measure or prediction quality.

Since the number of queries in Tweets2013 is relatively small, we expect it cannot support both tuning of predictors parameters and learning the regression model. Thus, we only consider Tweets2011. As discussed in Section 4.1.4, our earlier results on Tweets2011

were based on using 75% of the query set for parameters' tuning. If we follow such evaluation approach along with combining predictors, only 25% of the queries will be used in learning and testing the combined model. Thus, we consider a different evaluation setup for this task.

We adopt a similar evaluation setup to that discussed in Section 4.1.4, however we randomly split the query set using 40-60 splitting ratio, where we use 40% of the queries for parameters tuning, and the remaining 60% are used for learning and testing the linear regression model. This split-tune-test approach was repeated to generate 120 randomly-split query subsets. Once predictors are tuned for all queries in a subset, predictors are computed over the remaining 60% of the queries. Predicted values for a query can then be handled as features for that query. Thus, each query will be characterized by 73 features. The regression model is evaluated using 10-fold cross-validation over the 60% split of the queries. Model evaluation was repeated 120 times, and the quality of combining predictors will be the average of Pearson's correlation coefficient values over the 120 trials.

4.7.2 Feature Selection

Feature selection can help improve building the combined prediction model for several reasons. One example is related to the fact that some of the predictors we tested are correlated [3]. In such a case, considering all correlated predictors can be redundant which will not provide very different information in the learning process. To perform feature selection, we adopted a 2-step greedy approach that is a variant of the Greedy Stepwise approach [59].

In the first step, the method uses *forward addition* optimizing the average correlation to select the features subset to use in learning. The addition process navigates through the list of predictors ranked descendingly based on their individual prediction quality. In each addition step, the methods incrementally combines predictors, adding one at a time, and computes the average correlation of the newly formed set. The method continues till it reaches the maximal set of features including all predictors. All subsets are then ranked descendingly based on their prediction quality and the subset with the maximum performance is passed to the next step.

Starting by the optimal combined set, the algorithm then proceeds by following a backward elimination approach [59]. In each step, a different predictor is removed and the prediction quality of the combination of the remaining predictors is computed. Once the subset with the best performance is found, this elimination process is repeated on the new subset. The feature selection process stops once no improvement in prediction quality is achieved by eliminating any of the predictors. We have experimented with other feature selection methods and found this method to yield the best prediction quality.

4.7.3 Results and Discussion

In the following section, we present the Pearsons correlation coefficient values for the best individual predictor and combination of predictors for each retrieval model. Note that quality of individual predictors reported in this section is different from that reported

earlier due to the different evaluation setup followed. Table 4.14 presents a comparison between the quality of individual predictors and the best combination.

Table 4.14: Pearson's correlation values for best individual predictors and combined predictors for each retrieval model. Quality of combined set with significant improvement over individual predictor is marked with *, $p < 0.05$

Model	Best individual	Name	Combined	% improvment	Subset
QL	0.5237	MedLIdfTTC	0.5919*	13.02	{MeanTTC, CLR}
***t*-EXP**	0.4922	LowLIdfTTC	0.5875*	19.36	{LowLIdfTTC, MeanTTC, LowIdfTTC, t-CLR}
QE	0.4771	MeanLIdfTTC	0.5480*	14.86	{MeanLIdfTTC, Max-, Low-, Range-LIdfQTC}
***t*-QRM**	0.4907	UpLIdfTTC	0.5193	5.82	{UpLIdfTTC, Mean-LIdfQTC}

As the table shows, combining predictors resulted in a significant improvement over individual predictors for all models but t-QRM. With all remaining models, percentage of improvement exceeded 13% reaching a maximum of 20% with t-EXP. We argue that the low performance of combined predictors with t-QRM compared to other models is related to the feature selection approach we followed. In the first step of feature selection, the method incrementally add predictors sorted by their prediction quality and the subset with the maximum quality is kept for the second selection step. With t-QRM, UpLIdfTTC and MeanLIdfQTC where the best two predictors and when combined, they resulted in the best performing combination. Keep in mind that this combination will not necessarily be the *ultimate* best if all possible combinations of predictors were tested over the full predictors list. The fact that our feature selection method compromises running an exhaustive search over all feature space can result in such situation were the feature selection method stops following one pass of forward selection only.

We can also notice in the table that predictors based on our proposed variants are the best performing individual predictors and among the predictors in the combined subsets in all models but QL. It is also interesting to observe that microblog-specific predictors are generally the best performing individual predictors and also dominate combined subsets in all models.

Interestingly, we see that the temporal predictor t-CLR was among predictors in the best performing combination with t-EXP, supporting our previously made conclusion that a temporal predictor better fits a temporal retrieval model. A more stronger argument is summarized as follows. t-CLR was at rank 10 in the ranked predictors' list for the t-EXP model. It was eventually in the best combination resulting from forward addition feature selection step. Although preceded by stronger individual predictors, it survived all seven backward elimination passes, indicating that it had a crucial role in prediction with this model.

We see that CLR and t-CLR were among the best combinations in both QL and t-EXP respectively. In fact, omitting these clarity-based predictors from the best combination had a severe consequence on prediction performance as shown in Figure 4.24.

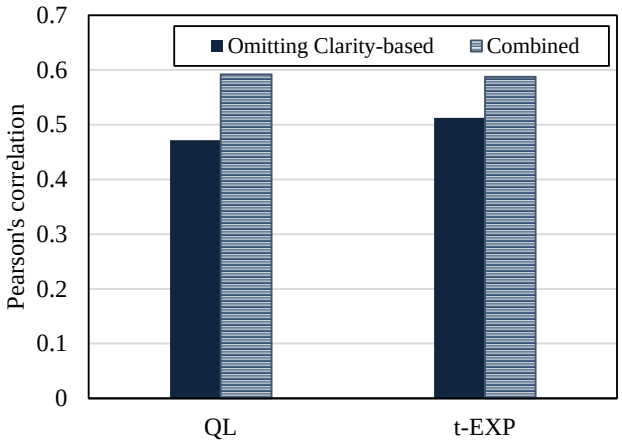


Figure 4.24: Pearson's correlation values for best combination of predictors before and after removing Clarity-based predictors.

The figure shows that omitting CLR and t-CLR from the best combinations for QL and t-EXP respectively, resulted in a significant drop in performance. This indicates that these predictors managed to cover aspects of query performance that were not covered by microblog-specific predictors. Further analysis is needed to understand this observation.

With the last set of results presented, we conclude the evaluation chapter and summarize the main conclusions we got in the following chapter.

Chapter 5

Conclusion and Future Work

Given the extensive experiments we performed in this first, large-scale study of QPP in context of microblog search, we came to some main conclusions and guidelines for future work that we discuss next.

5.1 Conclusion

We have experimented with a total of 73 predictors; 37 of them were state-of-the-art predictors and the remaining are based on variants we propose to some of the existing predictors. Predictors included pre- and post-retrieval ones, and temporal and non-temporal predictors. We tested prediction of two retrieval effectiveness measures used in evaluating microblog search: average precision (AP) and P@30 over four retrieval models used in microblog search. This study was carried over the two most widely-used tweets collections: Tweets2011 and Tweets2013.

Overall, the main conclusions we make in this thesis are as follows:

- The performance of the existing state-of-the-art predictors in the context of microblog search lies in range of their reported performance in other domains.
- Predicting both AP and P@30 with a relatively good prediction quality in a robust way across different retrieval models, is possible in the context of microblog search.
- Several experiments on temporal predictors showed that a strong temporal predictor (such as t-CLR) might better fit a temporal retrieval model. However, considering *surface* temporality of tweets relying on the tweet timestamps only in isolation from other relevant information might not be enough to produce good prediction in such temporal domain.
- It is possible to further improve prediction quality in this context. Our experiments on the variants we proposed show that they were generally the best in predicting AP and P@30 across different retrieval models and with different collections.
- Combining predictors also showed promising improvements over individual predictors achieving 13 to 20% improvement with almost all models.

- The performance of some of the existing predictors (e.g. CLR and WIG) was not consistent across different retrieval models. This requires further investigation and might mean that some predictors need to be re-designed to fit different retrieval models.

5.2 Future Work

Starting with this comprehensive study of QPP in microblog search, we develop several directions for future work. First, the study triggered the need for performance predictors that explicitly consider the temporal aspect of the task and the data. The new predictors might also leverage some specific features of the data, e.g., retweets and hashtags. Moreover, proposed predictors in this context can be designed to accommodate the specific nature of tweets including the very short length of text and the informality of the language usually used.

Second, the promising results of combining predictors indicate that this approach should be further investigated considering other learning schemes and feature selection methods.

Third, with the coming release of a new set of queries for Tweets2013[1], more extensive investigation of QPP with Tweets2013 is an interesting step ahead and can lead to different conclusions. We believe that predicting the performance of a larger set of queries with the very large number of tweets in Tweets2013 might reflect new insights on QPP in the current context.

Finally, using performance predictors in applications to improve microblog search effectiveness is definitely an interesting future direction. Due to the effectiveness of query expansion models in this context, the main applications we consider are those supporting such retrieval models. For example, *selective* query expansion can benefit from QPP to decide which queries should be expanded. A more general approach targets applying QPP to perform *dynamic* query expansion where QPP can help the search system in dynamically deciding the amount of expansion to apply for a given query. Additionally, we believe that the values of different predictors can be used as features in learning-to-rank-based retrieval, which is another interesting application to work on.

[1] https://github.com/lintool/twitter-tools/wiki/TREC-2014-Track-Guidelines

Bibliography

[1] Giambattista Amati, Claudio Carpineto, Giovanni Romano, and Fondazione Ugo Bordoni. Query difficulty, robustness and selective application of query expansion. In *eds, European Conf. on IR Research*, pages 127–137. Springer Berlin Heidelberg, 2004.

[2] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, October 2002.

[3] David Carmel and Elad Yom-Tov. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2(1):1–89, January 2010.

[4] Jaeho Choi and W. Bruce Croft. Temporal models for microblogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2491–2494. ACM, 2012.

[5] David M. Corey, William P. Dunlap, and Michael J. Burke. Averaging correlations: Expected values and bias in combined pearson rs and fisher's z transformations. *The Journal of General Psychology*, 125(3):245–261, July 1998.

[6] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, 2002.

[7] Steve Cronen-townsend, Yun Zhou, and W. Bruce Croft. A language modeling framework for selective query expansion. Technical Report IR-338, University of Massachusetts, Center for Intelligent Information Retrieval, 2004.

[8] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Precision prediction based on ranked list coherence. *Information Retrieval*, 9(6):723–755, December 2006.

[9] Ronan Cummins, Joemon Jose, and Colm O'Riordan. Improved query performance prediction using standard deviation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, 2011.

[10] Fernando Diaz. Performance prediction using spatial autocorrelation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 583–590. ACM, 2007.

[11] Fernando Diaz and Rosie Jones. Using temporal profiles of queries for precision prediction. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 18–24. ACM, 2004.

[12] Miles Efron. Query-specific recency ranking: Survival analysis for improved microblog retrieval. In *Proceedings of the 1st Workshop on Time-aware Information Access (#TAIA2012)*, TAIA '12, 2012.

[13] Miles Efron and Gene Golovchinsky. Estimation methods for ranking recent information. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, 2011.

[14] Miles Efron, Jimmy Lin, Jiyin He, and Arjen de Vries. Temporal feedback for tweet search with non-parametric density estimation. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 3342. ACM, 2014.

[15] Miles Efron, Peter Organisciak, and Katrina Fenlon. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 911–920. ACM, 2012.

[16] R. A. Fisher. *Statistical Methods For Research Workers*. Oliver & Boyd, 13 edition, 1958.

[17] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[18] Maram Hasanain, Rana Malhas, and Tamer Elsayed. Query performance prediction for microblog search: A preliminary study. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*, SoMeRA '14, pages 1–6. ACM, 2014.

[19] Claudia Hauff. *Predicting the Effectiveness of Queries and Retrieval Systems*. PhD thesis, University of Twente, The Netherlands, 2010.

[20] Claudia Hauff, Leif Azzopardi, and Djoerd Hiemstra. The combination and evaluation of query performance prediction methods. In *Advances in Information Retrieval*, number 5478 in Lecture Notes in Computer Science, pages 301–312. Springer Berlin Heidelberg, 2009.

[21] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 1419–1420. ACM, 2008.

[22] Claudia Hauff, Vanessa Murdock, and Ricardo Baeza-Yates. Improved query difficulty prediction for the web. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 439–448. ACM, 2008.

[23] Ben He and Iadh Ounis. Inferring query performance using pre-retrieval predictors. In *String Processing and Information Retrieval*, number 3246 in Lecture Notes in Computer Science, pages 43–54. Springer Berlin Heidelberg, 2004.

[24] Ben He and Iadh Ounis. Combining fields for query expansion and adaptive query expansion. *Information Processing & Management*, 43(5):1294–1307, September 2007.

[25] Rosie Jones and Fernando Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3):14:1–14:31, July 2007.

[26] Mostafa Keikha, Shima Gerani, and Fabio Crestani. Time-based relevance models. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11. ACM, 2011.

[27] Eyal Krikon, David Carmel, and Oren Kurland. Predicting the performance of passage retrieval for question answering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2451–2454. ACM, 2012.

[28] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, pages 79–86, 1951.

[29] Giridhar Kumaran and Vitor R. Carvalho. Reducing long queries using query quality predictors. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 564–571. ACM, 2009.

[30] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600. ACM, 2010.

[31] K. L. Kwok. A new method of weighting query terms for ad-hoc retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 187–195. ACM, 1996.

[32] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 111–119. ACM, 2001.

[33] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127. ACM, 2001.

[34] Xiaoyan Li and W. Bruce Croft. Time-based language models. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, CIKM'03, pages 469–475. ACM, 2003.

[35] Jimmy Lin and Miles Efron. Overview of the TREC-2013 Microblog Track. 2013.

[36] Jimmy Lin and Miles Efron. Temporal relevance profiles for tweet search. In *Proceedings of the 2nd Workshop on Time-aware Information Access (#TAIA2013)*, TAIA '13, 2013.

[37] Yang Liu, Ruihua Song, Yu Chen, Jian-Yun Nie, and Ji-Rong Wen. Adaptive query suggestion for difficult queries. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 15–24. ACM, 2012.

[38] Yuanhua Lv and ChengXiang Zhai. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 255–264. ACM, 2009.

[39] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, Cambridge, United Kingdom, 2008.

[40] Kamran Massoudi, Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In Paul Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, editors, *Advances in Information Retrieval*, number 6611 in Lecture Notes in Computer Science, pages 362–367. Springer Berlin Heidelberg, January 2011.

[41] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 472479. ACM, 2005.

[42] Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara. Combining recency and topic-dependent temporal variation for microblog search. In Pavel Serdyukov, Pavel Braslavski, Sergei O. Kuznetsov, Jaap Kamps, Stefan Rger, Eugene Agichtein, Ilya Segalovich, and Emine Yilmaz, editors, *Advances in Information Retrieval*, number 7814 in Lecture Notes in Computer Science, pages 331–343. Springer Berlin Heidelberg, January 2013.

[43] Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara. Improving pseudo-relevance feedback via tweet selection. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 439–448. ACM, 2013.

[44] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the TREC-2011 Microblog Track. 2011.

[45] Vassilis Plachouras, Ben He, and Iadh Ounis. University of glasgow at TREC 2004: Experiments in web, robust, and terabyte tracks with terrier. 2004.

[46] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference*

on Research and Development in Information Retrieval, SIGIR '98, pages 275–281. ACM, 1998.

[47] Joaqun Prez-Iglesias and Lourdes Araujo. Ranking list dispersion as a query performance predictor. In Leif Azzopardi, Gabriella Kazai, Stephen Robertson, Stefan Rger, Milad Shokouhi, Dawei Song, and Emine Yilmaz, editors, *Advances in Information Retrieval Theory*, number 5766 in Lecture Notes in Computer Science, pages 371–374. Springer Berlin Heidelberg, January 2009.

[48] Fiana Raiber and Oren Kurland. Using document-quality measures to predict web-search effectiveness. In *Advances in Information Retrieval*, number 7814 in Lecture Notes in Computer Science, pages 134–145. Springer Berlin Heidelberg, January 2013.

[49] Hadas Raviv, Oren Kurland, and David Carmel. Query performance prediction for entity retrieval. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 10991102. ACM, 2014.

[50] Jesus A. Rodriguez Perez and Joemon M. Jose. Predicting query performance in microblog retrieval. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 11831186. ACM, 2014.

[51] Mark Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, 2010.

[52] Anna Shtok, Oren Kurland, and David Carmel. Predicting query performance by query-drift estimation. In Leif Azzopardi, Gabriella Kazai, Stephen Robertson, Stefan Rger, Milad Shokouhi, Dawei Song, and Emine Yilmaz, editors, *Advances in Information Retrieval Theory*, number 5766 in Lecture Notes in Computer Science, pages 305–312. Springer Berlin Heidelberg, 2009.

[53] Anna Shtok, Oren Kurland, and David Carmel. Using statistical decision theory and relevance models for query-performance prediction. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 259–266. ACM, 2010.

[54] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. Predicting query performance by query-drift estimation. *ACM Trans. Inf. Syst.*, 30(2):11:1–11:35, 2012.

[55] Ian Soboroff, Iadh Ounis, Craig Macdonald, and Jimmy Lin. Overview of the TREC-2012 Microblog Track. 2012.

[56] Mor Sondak, Anna Shtok, and Oren Kurland. Estimating query representativeness for query-performance prediction. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 853–856. ACM, 2013.

[57] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. #TwitterSearch: A comparison of microblog search and web search. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 35–44. ACM, 2011.

[58] Stewart Whiting, Iraklis A. Klampanos, and Joemon M. Jose. Temporal pseudo-relevance feedback in microblog retrieval. In Ricardo Baeza-Yates, Arjen P. de Vries, Hugo Zaragoza, B. Barla Cambazoglu, Vanessa Murdock, Ronny Lempel, and Fabrizio Silvestri, editors, *Advances in Information Retrieval*, number 7224 in Lecture Notes in Computer Science, pages 522–526. Springer Berlin Heidelberg, January 2012.

[59] Ian H Witten, Eibe Frank, and Mark A Hall. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, third edition, 2011.

[60] Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 512–519. ACM, 2005.

[61] Ying Zhao, Falk Scholer, and Yohannes Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 52–64. 2008.

[62] Yun Zhou. *Retrieval performance prediction and document quality*. PhD thesis, University of Massachusetts Amherst, 2007.

[63] Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07. ACM, 2007.